A MICROCOMPUTER BASED FINANCIAL RECORD MANAGEMENT SYSTEM FOR PRODUCTION AGRICULTURE

by

William Scott Watson

An Abstract

Of a thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the School of Applied Sciences and Technology Central Missouri State University

November 1982

Thesis supervisor: Associate Professor Richard Tabor

As profit margins for agricultural products become narrower, the successful farm operator must maximize the efficiency of their management techniques. While carefully maintained manual bookkeeping systems can help identify weaknesses in the farming enterprise, the labor intensive nature of these systems rarely insures the timeliness of the information they produce. The scope of this thesis is to interface existing farm management financial record bookkeeping systems with the ease of use, dependability, and fast processing speed of a microcomputer.

While no original bookkeeping principles were utilized in this software package, certain abilities were incorporated that are generally considered too time consuming to be practical with manual bookkeeping systems. These abilities include:

1. The ability to manage and report information for more than one agricultural enterprise.
2. The ability to manage and report information for more than one bank account.
3. The ability to manage and report information for more than one farm operator.
4. The ability to collate various bookkeeping information to produce financial end-of-year analysis.
5. A degree of user friendliness that allows untrained users to operate the system with confidence.
6. Only information important to the user is displayed on the video screen or lineprinter.
   Complex data and program manipulations are not apparent to the user in order to avoid confusion.

The package was written using Disk BASIC computer language on a 48K Model I TRS-80 microcomputer with two five and one quarter inch mini-floppy diskette drives and one parallel interface eighty column lineprinter.

Abstract approved: Richard Tabor, thesis advisor

Chair, Ag Dept., title and department

11-1-82, date

CENTRAL MISSOURI STATE UNIVERSITY

WARRENSBURG, MISSOURI

This Thesis for the Master of Science Degree

by

William Scott Watson

Has been Approved for the Department
of
Agriculture

by

Richard Tabor
11-1-82

November 1982

A MICROCOMPUTER BASED FINANCIAL RECORD MANAGEMENT SYSTEM FOR PRODUCTION AGRICULTURE

A Thesis

Presented to

the Faculty of the School of Applied Sciences and Technology

Central Missouri State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

William Scott Watson

November 1982

Table Of Contents

Acknowledgments

The author wishes to express gratitude for the generosity of time and wisdom from the following persons:

John White, president, Computer Analysts

John O'Hara

Tom Lewis

Dr. Richard Tabor, Chairperson of Department of Agriculture, Central Missouri State University

John Watson

W. L. Watson

Joe Rowland, Department Head of Agriculture, Moniteau R-VI High School, Tipton Missouri

Vernon Hester, president, Cosmopoliton Electronics

Bill Lisica

Microprocessor And Peripheral Operation

The software presented in this thesis is designed to execute in a specific hardware environment. While translation of the software and substitution of the hardware is certainly possible, the following is a list of the necessary hardware to insure proper operation of the software as presented:

1. Radio Shack TRS-80 Model I or Model III microcomputer.
2. Parallel interfaced eighty column lineprinter.
3. Two double density, single sided, five and one quarter inch mini-floppy diskette drives.

The TRS-80 microcomputer utilizes a Zilog model Z-80 microprocessor as its central processing unit. The Z-80 has an eight bit data bus and a sixteen bit address bus, giving it the capacity to address a total of 65,536 eight-bit bytes. The native logic of all components of the TRS-80 is binary (or base two). All programs executed by the Z-80 must be encoded in memory as binary information.

Binary information, while inconvenient to humans as a number system, is essential to all microprocessor applications. In electronic format, the numbers one and zero are represented by the presence or non-presence of electricity at a given instant at a specific location. For purposes of nomenclature, a location which may contain a one or a zero electronically is termed a bit. By placing bits side-by-side, a larger number can be represented. In order to convert binary numbers to their decimal equivalents

it is necessary to understand how the binary number system operates.

As an example, suppose a four-bit binary number exists in an electronic circuit with a value of 1101_2 (the subscripted number implies the number base of the numerals; in this case, the subscripted '2' shows that 1101 is a binary number). In order to convert this value to decimal (base 10) equivalent, each bit must be converted to decimal individually. First, consider the decimal number 2,547. If this number is dissected one digit at a time from right to left we find:

$7 \times 10^0 = 7$

$4 \times 10^1 = 40$

$5 \times 10^2 = 500$

$2 \times 10^3 = 2{,}000$

Therefore: $2{,}000 + 500 + 40 + 7 = 2{,}547$

If this same process is executed on the binary number above:

$1 \times 2^0 = 1_{10}$

$0 \times 2^1 = 0_{10}$

$1 \times 2^2 = 4_{10}$

$1 \times 2^3 = 8_{10}$

Therefore: $8 + 4 + 0 + 1 = 13_{10}$

By observation, the largest number that four binary bits can represent is 1111_2 which would equal 15_{10}. As mentioned earlier, the Z-80 has a data bus of eight bits. The data bus width limits the largest number that can be passed to or from the central processing unit (or CPU) at one time. This places the numerical

range of the Z-80 at 0_{10} to 255_{10} (since 11111111_2 equals 255_{10}). The address bus determines how many memory locations can be accessed by the CPU for passing data. The sixteen bit wide address bus of the Z-80 allows up to 65,536 locations of memory to be accessible by the CPU. Again, as a matter of nomenclature, the number of bits assigned to a memory location is termed a byte. Since the data bus of the Z-80 is eight bits wide, it would be inefficient to use more or less than eight bits per byte. Thus, the TRS-80 uses eight-bit bytes as do nearly all other microcomputers. Progressive technology in the microprocessor field has produced a new generation of microprocessors which have a sixteen bit data bus and a thirty two bit address bus.

Memory can be defined as simply as a large collection of bits. It is up to the microprocessor to use the bits intelligently for storing data. In the TRS-80, memory is arranged in groups of eight bits, each group given an identifying number, called the address. Addresses are assigned in a sequential manner, beginning with memory location zero and ending with memory location 65,535. It is important to note here that memory is generally referred to in quantities of 'K'. While 'K' is the metric system abbreviation for one thousand, in computers 'K' is given the closest binary equivalent, 1024 (which is two to the tenth power). Therefore, we would refer to a computer having 65,536 bytes of memory (64 X 1024 bytes) as a '64K' computer.

Even though memory is given sequential addresses, the CPU does not have to search through the first 42,000 bytes to find the value of address 42,001. By simply placing the binary

equivalent on the address bus ($42,000_{10} = 1011011000010001_2$), the CPU can access the contents of that address directly. Because of this characteristic, this sort of memory is termed Direct Access Memory, or more commonly, Random Access Memory (RAM).

When power is first applied to the microprocessor, execution of an assumed program immediately begins at address zero. If no deliberate program instruction has been placed in this memory location the microprocessor will execute whatever random instruction does exist, and will continue to do so sequentially through memory indefinitely.

The Z-80 microprocessor has an instruction set of over 500 commands. While this may seem a large number, each command performs a somewhat primitive action, such as a branch in execution address, a simple addition, or a test of a single bit. Instructions exist as a series of one, two, three, or four bytes, depending on their relative complexity. For example, a single byte instruction is 00000000_2. This number refers to a specific instruction called No Operation. By the value of the first byte, the microprocessor knows if the instruction is a single byte instruction, or if further bytes need to be loaded to complete the instruction. Data is placed in memory the same way. Since the microprocessor cannot differentiate between program instruction bytes and data bytes, and error in program logic can cause disasterous results- causing the CPU to execute data values as if they were program instructions. Program instructions that can be processed directly by the microprocessor comprise an instruction set referred to as machine language.

Obviously, coding a machine language program by hand is a tedious and error-prone chore. The first high level language was written in order to solve this problem. In this language the individual instructions were given abbreviated labels. In the case of the No Operation instruction (00000000_2), the symbol NOP was assigned. Since these labels are easier to memorize and use than groups of binary digits, programming time was more efficiently used. Since the letters NOP are meaningless to a microprocessor, it was the duty of this language to translate these instruction symbols into their meaningful binary equivalents. This language was termed assembly language, and the actual program that did the translation was called the assembler.

Even though machine language instructions execute at incredible speeds (on the order of nanoseconds in the TRS-80), it is very difficult to write machine language programs because of the primitive nature of the instruction set and the inconvenient manner of numeric manipulation. The simple concept of floating point decimal numbers can take thousands of machine language instructions to emulate. Therefore, it became necessary to construct even higher level languages. Examples of these languages include FORTRAN, COBOL, and most recently, BASIC.

These languages became popular for several reasons. They were more English-like in their instruction sets, they performed difficult tasks with few commands, and they reduced the time necessary to train programmers. Even though these languages execute slower than pure machine language programs, it takes much less time to write the program and insure that it is operating correctly. Each of these languages is nothing more

than a computer program, designed to translate groups of symbols into machine language instructions.

Whenever electric power is applied to the TRS-80, a program called BASIC is immediately executed. This program is stored in a special non-volatile type of memory which has been addressed to the lowest 16K of memory. While random access memory loses its contents when power is turned off, BASIC is permanently burned into a type of memory called Read Only Memory (ROM). ROM does not lose its contents when power is turned off, and the contents of ROM cannot be changed once a program has been electrically burned into it. Thus, the TRS-80 operator does not have to learn complicated power-up procedures or key in BASIC by hand each time the computer is used. While it does limit the amount of memory that can be used for programs, it is a small trade for the convenience and power of a high level language such as BASIC.

It is for this reason that confusion exists about how much memory a fully expanded TRS-80 contains. A fully expanded TRS-80 is usually sold by the label '48K'. This refers only to the 48K bytes of memory available for user programs. Including the dedicated lower 16K of ROM, the computer actually contains a total of 64K bytes of memory.

RAM and ROM are not the only types of mass storage available to the microcomputer. Indeed, it would be unfortunate if a program would have to be retyped into the computer each time the power was disconnected. The most popular forms of mass storage for the TRS-80 are cassette tape and mini-floppy diskettes. An additional mass storage device would be a lineprinter. Even

though information cannot be reentered to the computer from a printout, it is a valid form of information storage.

The electronically simple processes of operating a cassette recorder and lineprinter are actually short enough to be included within the BASIC language. The more elaborate mini-floppy diskette drives require so many instructions to operate that a language separate from BASIC had to be written just for the use of the disk drives. When BASIC first begins to execute upon power-up, it electronically checks to see if a mini-floppy disk drive is connected to the computer. If it finds at least one disk drive connected it begins a special program sequence. If it does not find a disk drive connected, BASIC continues its power-up sequence as normal.

The special program sequence executed when a disk drive is detected is known as a boot sequence; a term derived from the phrase "picking oneself up by the bootstraps". During the boot sequence, execution of BASIC is aborted, the disk drive is turned on, and a special program called the Disk Operating System or DOS is loaded into the computer and executed. DOS (pronounced to rhyme with boss) contains several programs designed to fully utilize the capabilities of the mini-floppy disk drive.

In addition DOS contains a small program called Disk BASIC. Disk BASIC interfaces with the resident ROM BASIC adding several commands to use the mini-floppy disk drive to save and load programs as well as to facilitate storage and retrieval of data files. However, since at least part of DOS, and probably Disk BASIC will be in memory, the amount of user available memory is lowered to around 32K in a fully expanded system.

The method by which data is encoded on a mini-floppy diskette is of prime concern to the programmer. A mini-floppy diskette is divided into forty concentric circles, called tracks. Each track is divided into a variable number of pie-shaped wedges which are called sectors. For the TRS-80 mini-floppy diskette operating at double density, each track is divided into sixteen sectors of 256 bytes each. Therefore, a mini-floppy diskette would have the capacity of 163,840 bytes of storage. For a computer having only one disk drive part of that storage space will be dedicated to DOS. However, each additional disk drive would have the full 163,840 bytes available to the user as storage space.

One track of each diskette is reserved for the diskette's directory. The directory instructs DOS where on the disk certain programs or files are located. In addition, the directory maintains a table of free space available, and where it is located on the diskette. For this table, each track is split in half and that group of eight sectors is called a granule. The free space map keeps track of each granule's status- whether it is being used or not. If a data file containing only seven bytes of data were to be stored on the diskette, an entire granule of 2,048 bytes would still be allocated to the file.

In order to operate properly, the software presented in this thesis relies upon the above hardware environment. Since all programs are written using Disk BASIC, it is not recommended that any of the above hardware specifications be changed without full understanding of the software.

DOS Operation

From a hardware view, microcomputers may seem an intricate world of binary and acronyms. However, the ultimate goal of a sound computer program is to be easily understood and used by persons untrained in microprocessor and peripheral operation. Again, it is the duty of the programmer to utilize the computer's power to simplify the task of the end user. Since the software presented in this thesis heavily depends on fast, error free disk operation, a short discussion of disk software operation is presented here.

As mentioned earlier, a mini-floppy diskette may contain both program or data files. The type of file on the diskette is identified by information in the diskette's directory. Each file, whether it is a program or data file, is given a descriptive name. This name may be up to eight characters in length, and may contain both letters and numbers as long as the first character is a letter. In addition, the program name is given an extension which may be up to three characters in length. An extension tells the disk user what type of file it is (program or data).

In the disk directory, the program name and extension are separated by a backslash ('/') character. It is left to the programmer to be consistent in extension names to avoid confusion at a later date. There are three uniform extensions used by the software in this thesis:

1. BAS - identifies file as a BASIC program.
2. CMD - identifies file as a machine language program.
3. DAT - identifies file as a data file.

Therefore, a directory entry called EDITOR/BAS would show there is a file on this diskette called EDITOR and it is a BASIC program. Likewise, an entry called NAMES/DAT would show there is a file called NAMES and it is a sequential data file. Athough several other conventional extension names exist, these are the only ones used by the programs in this thesis.

There are only three Disk BASIC commands which govern the use of BASIC programs: LOAD, RUN, and SAVE. Each of these commands require a target file specification placed in quotes, such as the command LOAD "EDITOR/BAS". In addition, a disk drive number can be specified at the end of the file specification to speed up searching for the file. An example of this is: LOAD "EDITOR/BAS:0".

The SAVE "FILESPEC" command would magnetically write on the diskette contained in the specified drive (or the first drive found to have free space if no drive number is specified) whatever BASIC program is in memory at that time. The LOAD "FILESPEC" command would retrieve a previously stored program and place it into memory, ready to execute. The RUN "FILESPEC" command functions exactly as the LOAD "FILESPEC" command, but execution of the program begins automatically after it has been loaded into memory. Using just these three commands, all BASIC program manipulation is accomplished.

Data file manipulation is not quite as straightforward as BASIC program manipulation. The sequential data file has a first-in-first-out (FIFO) access arrangement. That is, to read the fifty first data record in a sequential data file, the first

fifty records would have to be read in and discarded. The file is made accessable to the BASIC program by a command called OPEN. The OPEN command must specify whether the data file is to be input from or output to because DOS will not allow data to be moving in opposite directions simultaneously. In order to change a file access from input to output, or vice versa, the file must be closed using the command CLOSE and then reopened with the desired specification of data movement direction.

The largest limitation of the sequential data file is the inability to easily edit mistakes from the file. Since information cannot be inserted or deleted from the middle of a data file, a copy of the file must be made with the corrections placed in the copy. Then the old file is destroyed and the new file is renamed the old file's directory name. Fortunately, DOS does allow new data to be added to the end of a sequential data file.

The nature of sequential data files defines its use for data that once entered does not change. A permanent file of receipts or check stubs, as used by the programs in this thesis, would meet this criterion.

Program Overview

The programs presented in this thesis require a diskette storage space of approximately 125,000 bytes, and an execution memory (RAM) of nearly 250,000 bytes. Since this is more memory than is available on a fully expanded (48K RAM) TRS-80, a chained module method of execution was incorporated.

The chained module method splits the overall function of the package into its integral parts. A separate program is used to maintain each of these parts, and a menu driven selection scheme allows the user to select the appropriate program for the desired task. When a particular task is completed, the menu program is executed, and the user can select the next desired task. An important consideration for the use of chained program modules is not to allow any single program to occupy more memory than is available.

Two disk drives are used in the execution of this package. One drive is used to hold the program diskette, while the other is used for storage and manipulation of data files. Two small data files are kept on the program diskette, however, to identify the diskette and the current configuration of the package. This was necessary since the programs needed more storage space than is available on a single diskette, and were divided into two program diskettes. The identification file thus instructs the menu program which of the program diskettes is currently in the disk drive so that a diskette switch could be made if necessary.

The configuration file is a vital part of the package, its function being to customize the package to the individual requirements of the user. This file is accessed by nearly all of the programs in the package constantly, so the placement of the configuration file on the program disk was done to increase efficiency. The configuration file contains the codes and descriptions of all expense and receipt account categories, codes and descriptions of all enterprises, codes and descriptions of all bank accounts, and codes and descriptions of all farm operators.

Three types of data files are used by this system: receipts, expenses, and special. A different diskette is used for each type of file. Each of these diskettes has an identification file to instruct programs whether the correct data diskette is in the disk drive for the particular job being executed. This method is used to keep the various data separated and eliminate the possibility of incorrect data being stored on a data diskette. The special data diskette is used for all data not associated with receipts and expenses. Examples of this data would be inventories, capital gains records, and depreciation schedules.

For security, the identification files on both data and program diskettes are similar in structure. This eliminates the possibility of accidental placement of a program disk in the data drive, and having data written over important programs. Before any data is written to a diskette, its identification file is interrogated. If the wrong diskette is inserted, the user is alerted and given the opportunity to correct the mistake.

At any time, the user may abort a specific job and return to the menu. In addition, the identification file allows the data file to actually span more than one physical diskette, that is, a receipts file can be as long as the user needs. No limits are placed on the number of entries that can be made to a data file. When a certain number of entries have been stored in a data file, the user is instructed to initialize a new data diskette.

Program Operation

The scope of this thesis is the construction of a financial record management system for production agriculture implemented on a microcomputer. Actual information manipulation is no different in this system than in conventional manual bookkeeping systems. The particular bookkeeping system emulated in this system is the Ohio State University Commercial Farm Account Record Book. Because of the flexibility of the configuration file used by the computerized bookkeeping system, it is compatible with any manual bookkeeping system for agricultural use.

In addition to providing timely information by virtue of the computer's extremely fast processing abilities, the package was designed to include the following abiliites:

1. Ability to manage and report information for more than one agricultural enterprise.
2. Ability to manage and report information for more than one bank account.
3. Ability to manage and report information for more than one farm operator.
4. Ability to collate various information to provide end-of-year financial evaluation of any of the above criteria or the operation as a whole. This information includes debit and credit summary for each receipt and expense account category, depreciation schedules, and net worth statement.

5. A degree of user friendliness that allows untrained users to operate the system with confidence.
6. Complex data and program manipulations must not be apparent to the user to avoid confusion. Only information important to the user is displayed on the video screen or lineprinter.

The configuration file used by the system is defined by the user, as well as fiscal year information. This allows a user to adapt the system to the manual system that is most familiar to them. In addition, the user may expand the configuration to include features that manual systems do not employ, such as enterprise and bank account cross referencing reports. The ultimate goal of this project was to provide a record management system so flexible that a user could easily generate any cross reference of the data entered into the system. For instance, a valid report might be the amount of veterinary bills used for a swine operation which were paid by cash between the dates of February 23 and June 15.

Each of the program modules in the package perform a specific function. The following is a description of each of the programs included in the package.

ANALYSIS/BAS - This program surveys and collates all credit and debit information to produce a profit or loss statement.

BACK/BAS - This program provides a means to make backup copies of data diskettes- a necessary function of good data processing operation.

CGMEBF/BAS - This program stores and prints out a record of Capital Gains For Machinery, Equipment, Buildings, and Farm Land.

CGPBLSD/BAS - This program stores and prints out a record of Capital Gains For Purchased Breeding Livestock Sold Or Died.

CGRBL/BAS - This program stores and prints out a record of Capital Gains For Raised Breeding Livestock.

CONFIG/BAS - This is the largest program in the package due to its enormous flexibility. The function of this program is to configure the bookkeeping parameters based upon the preference of the user. From this information a file entitled CONFIG/DAT is placed on both program diskettes and is accessed by any program that performs a bookkeeping function. This program can be used to create CONFIG/DAT, add or delete items from CONFIG/DAT, print out a hardcopy record of CONFIG/DAT, sort CONFIG/DAT by code number, or sort CONFIG/DAT alphabetically

DEPREC/BAS - This program stores a permanent depreciation schedule for depreciable items on the farm. Although the program does not calculate actual depreciation schedules, the information is necessary to perform end-of-year analysis.

EDITOR/BAS - This progra allows the user to correct mistakes in a receipts or expenses data file. The program can display entries, delete entries, restore deleted entries, or purge a data diskette of deleted entries to make room for additional entries on that diskette.

EXPENSES/BAS - This program is used to enter and store expenses incurred. All expense entries are made to the EXPENSES data disk and the information file is updated to show the current number of expense entries in the file.

FINVEN/BAS - This program stores or prints out a record of Inventory of Feed, Crops, and Supplies. Like other inventories and capital gains records, this data is placed on the SPECIAL data diskette.

FISCAL/BAS - This program allows the user to customize the package to perform end-of-year analysis based on the fiscal year dates used by that farm operation. This data is placed in a file called FISCAL/DAT and stored on the SPECIAL data diskette.

INIT/BAS - This program allows the user to create a new RECEIPTS, EXPENSES, or SPECIAL data diskette. An identification file entitled HEADER/DAT contains either the word RECEIPTS, EXPENSES, or SPECIAL in order to alert bookkeeping programs which type of data diskette has been placed in a disk drive.

MASTER/BAS - This is the program which acts as a menu to select other programs in the package. Upon power-up, this program is automatically executed and upon completion of any other program this program is again loaded and executed. Unlike any other program in the package, this program is placed on both program disks.

MINVEN/BAS - This program stores and prints out a record of Inventory of Market Animals, Feeders, and Poultry.

NETWORTH/BAS - This program surveys all credit and debit data entered and calculates a net worth statement. A copy of the statement is retained on the SPECIAL data diskette for later reference.

RECEIPTS/BAS - This program is used to enter and store receipts obtained. All receipt entries are made to the RECEIPTS data diskette and the identification file is updated to show the current number of receipt entries in the file.

REPORTER/BAS - This program is used to construct and print out cross reference reports to the user's specification. As its function is only to survey and collate data, it performs no file manipulation or bookkeeping operations.

RINVEN/BAS - This program stores and prints out a record of Inventory of Raised Breeding Livestock On Hand At End Of Year.

SUMMARY/BAS - This program totals and prints out monthly and year end records of receipts and expenses by each receipt and expense account category. These totals are posted to a special file on the SPECIAL diskette to be used by other programs such as ANALYSIS/BAS and NETWORTH/BAS.

APPENDIX I: Program Listings

Program Listing: ANALYSIS/BAS

```
10 CLS:CLEAR 10000:FI$="$$#,###,###.##":PRINT">> Option:
Print Out A Financial Analysis of Year's Business
<<":PRINT:PRINT
20 PRINT"Do you wish to:":PRINT:PRINT"1) Print out
analysis":PRINT"2) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1 or 2)"
30 PO%=539:A1%=1:GOSUB40160:A$=AN$:IF A$<>"1" AND A$<>"2"
GOTO 30 ELSE IF A$="2" THEN CLS:PRINT"Returning to MASTER
MENU":RUN "MASTER/BAS"
40 CLS:PRINT"Place the disk marked SPECIAL in Drive 1, then
Press <ENTER>"
50 PO%=128:A1%=0:GOSUB40130
60 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:CLOSE:IF
D$="SPECIAL" GOTO 100 ELSE CLS:PRINT"ERROR - this is not a
SPECIAL disk!  It is marked: ";D$
70 PRINT:PRINT"Do you wish to:":PRINT:PRINT"1) Try
again":PRINT"2) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1 or 2)"
80 PO%=475:A1%=1:GOSUB40160:A$=AN$:IF A$<>"1" AND A$<>"2"
GOTO 80 ELSE IF A$="2" THEN CLS:PRINT"Returning to MASTER
MENU":RUN"MASTER/BAS"
90 GOTO 40
100 OPEN "I",1,"HEADER/DAT:1":DIM ST(14):INPUT#1,D$:FOR Z=1
TO 14:INPUT#1,ST(Z):NEXT Z:CLOSE
110 IF ST(1)<>0 AND ST(2)<>0 AND ST(3)<>0 AND ST(4)<>0 AND
```

```
ST(5)<>0 AND ST(6)<>0 AND ST(7)<>0 AND ST(8)<>0 GOTO 200
120 CLS:PRINT"ERROR - before I can proceed you must:":PRINT
130 C=0:FOR Z=1 TO 8:READ A$:IF ST(Z)<>0 THEN NEXT Z:GOTO
190
140 C=C+1:PRINTC;") ";A$:NEXT Z
150 DATA "Post a summary of year's business to disk","Post
an inventory of Feed, Crops, and Supplies ","Post an
inventory of Market Animals, Feeders, and Poultry "
160 DATA "Post an inventory of Raised Breeding Livestock on
hand","Post Capital Gains for Raised Breeding
Livestock","Post Capital Gains for Purchased Breeding
Livestock"
170 DATA "Post Capital Gains for Machinery, Equip., Bldgs.,
and Farm","Post Annual Depreciation Record"
190 PRINT@960,"Press <ENTER> to
continue";:PO%=1020:A1%=0:GOSUB 40130:CLS:PRINT"Returning to
MASTER MENU":RUN"MASTER/BAS"
200 CLS:PRINT"Retrieving Fiscal Year Data":DIM
BM(12),BD(12),M(12),D(12):OPEN "I",1,"FISCAL/DAT:1":FOR Z=1
TO 12:INPUT#1,BM(Z),BD(Z),M(Z),D(Z):NEXT Z:CLOSE
210 CLS:PRINT"Retrieving inventory of Feed, Crops, and
Supplies"
220 OPEN"i",1,"FINVEN/DAT:1"
230 INPUT#1,D$:IF D$="TOTALS" GOTO 240 ELSE
INPUT#1,QB$,QE$,PB#,PE#,TB#,TE#:GOTO 230
240 INPUT#1,TB#,TE#:CLOSE:C1#=TE#-TB#
250 CLS:PRINT"Retrieving inventory of Market Animals,
```

```
Feeders, & Poultry":OPEN "I",1,"MINVEN/DAT:1"
260 INPUT#1,D$:IF D$="TOTALS" GOTO 270 ELSE
INPUT#1,NB,NE,BW#,EW#,PB,PE,TB#,TE#:GOTO 260
270 INPUT#1,TB#,TE#:CLOSE:C2#=TE#-TB#
280 CLS:PRINT"Retrieving inventory of Raised Breeding
Livestock":OPEN "I",1,"RINVEN/dat:1"
290 INPUT#1,D$:IF D$="TOTALS" GOTO 300 ELSE
INPUT#1,BN,EN,BV#,EV#:GOTO 290
300 INPUT#1,BV#,EV#:CLOSE:C3#=EV#-BV#
310 CLS:PRINT"Retrieving Capital Gains for Raised Breeding
Livestock":OPEN "I",1,"crbl/dat:1"
320 G1#=0:FOR Z=1 TO
ST(5):INPUT#1,DA$,NS,TR#:G1#=G1#+TR#:NEXT Z:CLOSE
330 CLS:PRINT"Retrieving Capital Gains for Purchased
Breeding Livestock":OPEN"I",1,"cpblsd/dat:1":G2#=0:FOR Z=1
TO ST(6)
340 INPUT#1,DA$,NA,SV#,UC#,CG#,IC#:G2#=G2#+CG#:NEXT Z:CLOSE
350 CLS:PRINT"Retrieving Capital Gains for Machinery,
Equip., Bldgs., & Farm":OPEN "I",1,"cmebf/dat:1":G3#=0:FOR
Z=1 TO ST(7)
360 INPUT#1,DE$,DA$,NS#,DE#,OC#,CG#:G3#=G3#+CG#:NEXT Z:CLOSE
370 CLS:PRINT"Retrieving annual depreciation
record":OPEN"I",1,"deprec/dat:1":G4#=0:G5#=0:G6#=0:FOR Z=1
TO ST(8):INPUT#1,DE$,DA$,TY,UC#,DY#
380 IF TY=1 THEN G4#=G4#+DY# ELSE IF TY=2 THEN G5#=G5#+DY#
ELSE IF TY=3 THEN G6#=G6#+DY#
390 NEXT Z:CLOSE
```

```
400 CLS:PRINT"Retrieving annual summary of receipts and
expenses"
410 DIM R#(12),E#(12):OPEN "I",1,"summary/dat:1":FOR Z=1 TO
12:INPUT#1,R#(Z),E#(Z):NEXT Z:CLOSE
420 CLS:PRINT"All necessary information has been
retrieved.":PRINT"Ready printer, then Press <ENTER>"
430 PO%=128:A1%=0:GOSUB40130
440 A$="FINANCIAL ANALYSIS OF YEAR'S BUSINESS":LPRINT
TAB(33-(LEN(A$)/2));A$:LPRINT
TAB(33-(LEN(A$)/2));STRING$(LEN(A$),"="):LPRINT:LPRINT
450 LPRINT STRING$(66,"-"):LPRINT "Receipts and
Expenses";TAB(35);"GAIN";TAB(55);"LOSS":LPRINT
STRING$(21,"-");TAB(30);STRING$(14,"-");TAB(50);STRING$(14,"-
")
455 FOR Z=1 TO
12:BM$=STR$(BM(Z)):BD$=STR$(BD(Z)):M$=STR$(M(Z)):D$=STR$(D(Z)
):BM$=RIGHT$(BM$,LEN(BM$)-1):BD$=RIGHT$(BD$,LEN(BD$)-1):M$=RI
GHT$(M$,LEN(M$)-1):D$=RIGHT$(D$,LEN(D$)-1)
460 P$=BM$+"/"+BD$+" to "+M$+"/"+D$:LPRINT
P$;TAB(30);:LPRINTUSING FI$;R#(Z);:LPRINTTAB(50);:LPRINT
USING FI$;E#(Z):TG#=TG#+R#(Z):TL#=TL#+E#(Z):NEXTZ:LPRINT
STRING$(66,"-")
470 LPRINT "Total Cash Receipts";TAB(30);:LPRINT USING
FI$;TG#:LPRINT "Total Cash Expenses";TAB(50);:LPRINT USING
FI$;TL#:LPRINT STRING$(66,"-")
480 LPRINT "Change In
Inventory";TAB(35);"GAIN";TAB(55);"LOSS":LPRINT
```

```
STRING$(19,"-");TAB(30);STRING$(14,"-");TAB(50);STRING$(14,"-
")
490 LPRINT "Market Animals";:IF C2#<=0 THEN LPRINT
TAB(50);:C2#=-(C2#):TL#=TL#+C2#:ELSE LPRINT
TAB(30);:TG#=TG#+C2#
500 LPRINT USING FI$;C2#
510 LPRINT "Raised Breeding Stock";:IF C3#<=0 THEN LPRINT
TAB(50);:C3#=-(C3#):TL#=TL#+C3#:ELSE LPRINT
TAB(30);:TG#=TG#+C3#
520 LPRINT USING FI$;C3#
530 LPRINT "Feed, Crops, & Supplies";:IF C1#<=0 THEN LPRINT
TAB(50);:C1#=-(C1#):TL#=TL#+C1#:ELSE LPRINT
TAB(30);:TG#=TG#+C1#
540 LPRINT USING FI$;C1#
550 LPRINT STRING$(66,"-")
560 LPRINT "Capital Gains or
Losses";TAB(35);"GAIN";TAB(55);"LOSS":LPRINT
STRING$(23,"-");TAB(30);STRING$(14,"-");TAB(50);STRING$(14,"-
")
570 LPRINT "Raised Breeding Stock";TAB(30);:LPRINT USING
FI$;G1#:TG#=TG#+G1#
580 LPRINT "Purchased Breeding Stock";:IF G2#<=0 THEN LPRINT
TAB(50);:G2#=-(G2#):TL#=TL#+G2#:ELSE LPRINT
TAB(30);:TG#=TG#+G2#
590 LPRINT USING FI$;G2#
600 LPRINT "Machinery & Equipment";:IF G3#<=0 THEN LPRINT
TAB(50);:G3#=-(G3#):TL#=TL#+G3#:ELSE LPRINT
```

```
TAB(30);:TG#=TG#+G3#
610 LPRINT USING FI$;G3#
620 LPRINT STRING$(66,"-"):LPRINT
"DePreciation";TAB(35);"GAIN";TAB(55);"LOSS":LPRINT
STRING$(12,"-");TAB(30);STRING$(14,"-");TAB(50);STRING$(14,"-
")
630 LPRINT "Buildings & ImProvements";TAB(50);:LPRINT USING
FI$;G4#:TL#=TL#+G4#
640 LPRINT "Machinery & EquiPment";TAB(50);:LPRINT USING
FI$;G5#:TL#=TL#+G5#
650 LPRINT "Purchased Breeding Stock";TAB(50);:LPRINT USING
FI$;G6#:TL#=TL#+G6#
660 LPRINT STRING$(66,"-"):LPRINT
STRING$(66,"-"):LPRINT"Total Gains";TAB(30);:LPRINT USING
FI$;TG#
670 LPRINT "Total Losses";TAB(50);:LPRINT USING
FI$;TL#:LPRINT STRING$(66,"-"):LPRINT STRING$(66,"-")
680 ID#=TG#-TL#:IF ID#<=0 THEN LPRINT "Total Farm
Loss";TAB(30);:ELSE LPRINT "Total Farm Profit";TAB(30);
690 LPRINT USING FI$;ID#:LPRINT STRING$(66,"-"):LPRINT
STRING$(66,"-")
700 CLS:PRINT"Returning to MASTER MENU":RUN "MASTER/BAS"
40070 AN$=" ":POKE VARPTR(AN$),A1%:POKE
VARPTR(AN$)+2,INT(PO%/256)+60:POKE
VARPTR(AN$)+1,PO%-INT(PO%/256)*256:RETURN
40130 A%=0:PRINT@PO%,STRING$(A1%,132);
40131 IF A%=A1%THEN40134ELSEPRINT@PO%+A%,CHR$(132);
```

```
40132 A$=INKEY$:IFA$=""THEN40132ELSEIFINSTR("
!#$%&'()*+<-.?0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZabcd
efghijklmnopqrstuvwxyz",A$)THENPRINT@PO%+A%,A$;:A%=A%+1:GOTO4
0131
40133
ONINSTR(CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40135,40130,40138:GO
TO40131
40134 A$=INKEY$:IF A$=""THEN40134ELSE40133
40135 IFA%<A1%THENPRINT@PO%+A%,CHR$(132);
40136 A%=A%-1:IFA%<0THENA%=0:GOTO40131ELSE40131
40137 A%=0
40138
IFA$=CHR$(91)THENPRINT@PO%,STRING$(A1%,132);ELSEPRINT@PO%+A%,
STRING$(A1%-A%," ");
40139 GOSUB40070:RETURN
40140 S%=1:AN$="":PRINT@PO%,"$";STRING$(A1%,132);"
";:PRINT@PO%+A1%-2,".";
40141
A$=INKEY$:IFA$=""THEN40141ELSEIFINSTR("0123456789",A$)THEN401
43ELSEONINSTR("-"+CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40145,4014
0,40140,40147
40142 GOTO40141
40143
AN$=AN$+A$:IFLEN(AN$)=1THENAN$=CHR$(132)+AN$ELSEIFLEN(AN$)>A1
%-1THENAN$=LEFT$(AN$,A1%-1)ELSEIFLEN(AN$)=3ANDLEFT$(AN$,1)=CH
R$(132)THENAN$=RIGHT$(AN$,2)
40144
```

```
PRINT@PO%+A1%-LEN(AN$),LEFT$(AN$,LEN(AN$)-2);".";RIGHT$(AN$,2
);:GOTO40141
40145
S%=-S%:PRINT@PO%+A1%+1,"";:IFS%=-1THENPRINT"-";:GOTO40141ELSE
PRINT" ";:GOTO40141
40146 IF A$=CHR$(91)THENPRINT@PO%+1,STRING$(A1%,132);"
";:PRINT@PO%+A1%-2,".";:GOTO40149ELSEPRINT@PO%,STRING$(A1%+2,
" ");:GOTO40149
40147 IFLEN(AN$)=0THENPRINT@PO%,STRING$(A1%+2,"
");:GOTO40149 ELSEPRINT@PO%+1,STRING$(A1%-1-LEN(AN$),"
");:IFLEFT$(AN$,1)=CHR$(132)THENPRINT@PO%+A1%-1,"0";:MID$(AN$
,1,1)="0"
40148
AN$=MID$(AN$,1,LEN(AN$)-2)+"."+RIGHT$(AN$,2):IFS%=-1THENAN$="
-"+AN$
40149 GOSUB 40170:RETURN
40160 S%=1:AN$="":PRINT@PO%,STRING$(A1%,132);" ";
40161
A$=INKEY$:IFA$=""THEN40161ELSEIFINSTR("0123456789",A$)THEN401
62ELSEONINSTR(CHR$(8)+CHR$(31)+"."+"-"+CHR$(13),A$)GOTO40160,
40160,40165,40163,40166:GOTO40161
40162
AN$=AN$+A$:IFLEN(AN$)>A1%THENAN$=LEFT$(AN$,A1%):GOTO40161ELSE
PRINT@PO%+A1%-LEN(AN$),AN$;:GOTO40161
40163
S%=-S%:PRINT@PO%+A1%,"";:IFS%=-1THENPRINT"-";ELSEPRINT" ";
40164 GOTO40161
```

```
40165 IFINSTR(AN$,".")=0THEN40162ELSE40161
40166 IFAN$=""THEN40168ELSEPRINT@PO%,STRING$(A1%-LEN(AN$),"
");
40167 IFS%=-1THENAN$="-"+AN$:GOTO40169ELSE40169
40168 IF A$=CHR$(91)THENPRINT@PO%,STRING$(A1%,132);"
";ELSEPRINT@PO%,STRING$(A1%," ");" ";
40169 GOSUB 40170:RETURN
40170 AN#=VAL(AN$):AN#=(INT((AN#+.005)*100)/100):RETURN
50000 REM * Data Entry Subroutines                          *
50001 REM * Gosub 40140 - Dollar entry                      *
50002 REM * Gosub 40150 - Numeric entry                     *
50003 REM * Gosub 40130 - Alpha/numeric entry               *
50004 REM * Parameters:                                     *
50005 REM *    Input:  PO% = Print@ position                *
50006 REM *            A1% = Number of characters to allow*
50007 REM *    Output: AN$ = Response                       *
50008 REM *            AN# = Numeric rounded equivalent    *
```

Program Listing: BACK/BAS

```
10 CLS:PRINT">> Option: Backup A Data Diskette
<<":PRINT:PRINT"Do You Wish To:":PRINT:PRINT"1) Execute
Backup":PRINT"2) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1 or 2)"
20 PO%=475:A1%=1:GOSUB40160:A$=AN$:IF A$<>"1" AND A$<>"2"
GOTO 20 ELSE IF A$="2" THEN CLS:PRINT"Returning to MASTER
MENU":RUN "MASTER/BAS"
30 CMD "backup"
40 RUN
40070 AN$=" ":POKE VARPTR(AN$),A1%:POKE
VARPTR(AN$)+2,INT(PO%/256)+60:POKE
VARPTR(AN$)+1,PO%-INT(PO%/256)*256:RETURN
40130 A%=0:PRINT@PO%,STRING$(A1%,132);
40131 IF A%=A1%THEN40134ELSEPRINT@PO%+A%,CHR$(132);
40132 A$=INKEY$:IFA$=""THEN40132ELSEIFINSTR("
!#$%&'()*+<-.?0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZabcd
efghijklmnopqrstuvwxyz",A$)THENPRINT@PO%+A%,A$;:A%=A%+1:GOTO4
0131
40133
ONINSTR(CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40135,40130,40138:GO
TO40131
40134 A$=INKEY$:IF A$=""THEN40134ELSE40133
40135 IFA%<A1%THENPRINT@PO%+A%,CHR$(132);
40136 A%=A%-1:IFA%<0THENA%=0:GOTO40131ELSE40131
40137 A%=0
```

```
40138
IFA$=CHR$(91)THENPRINT@PO%,STRING$(A1%,132);ELSEPRINT@PO%+A%,
STRING$(A1%-A%," ");
40139 GOSUB40070:RETURN
40140 S%=1:AN$="":PRINT@PO%,"$";STRING$(A1%,132);"
";:PRINT@PO%+A1%-2,".";
40141
A$=INKEY$:IFA$=""THEN40141ELSEIFINSTR("0123456789",A$)THEN401
43ELSEONINSTR("-"+CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40145,4014
0,40140,40147
40142 GOTO40141
40143
AN$=AN$+A$:IFLEN(AN$)=1THENAN$=CHR$(132)+AN$ELSEIFLEN(AN$)>A1
%-1THENAN$=LEFT$(AN$,A1%-1)ELSEIFLEN(AN$)=3ANDLEFT$(AN$,1)=CH
R$(132)THENAN$=RIGHT$(AN$,2)
40144
PRINT@PO%+A1%-LEN(AN$),LEFT$(AN$,LEN(AN$)-2);".";RIGHT$(AN$,2
);:GOTO40141
40145
S%=-S%:PRINT@PO%+A1%+1,"";:IFS%=-1THENPRINT"-";:GOTO40141ELSE
PRINT" ";:GOTO40141
40146 IF A$=CHR$(91)THENPRINT@PO%+1,STRING$(A1%,132);"
";:PRINT@PO%+A1%-2,".";:GOTO40149ELSEPRINT@PO%,STRING$(A1%+2,
" ");:GOTO40149
40147 IFLEN(AN$)=0THENPRINT@PO%,STRING$(A1%+2,"
");:GOTO40149 ELSEPRINT@PO%+1,STRING$(A1%-1-LEN(AN$),"
");:IFLEFT$(AN$,1)=CHR$(132)THENPRINT@PO%+A1%-1,"0";:MID$(AN$
```

```
,1,1)="0"
40148
AN$=MID$(AN$,1,LEN(AN$)-2)+"."+RIGHT$(AN$,2):IFS%=-1THENAN$="
-"+AN$
40149 GOSUB 40170:RETURN
40160 S%=1:AN$="":PRINT@PO%,STRING$(A1%,132);" ";
40161
A$=INKEY$:IFA$=""THEN40161ELSEIFINSTR("0123456789",A$)THEN401
62ELSEONINSTR(CHR$(8)+CHR$(31)+"."+"-"+CHR$(13),A$)GOTO40160,
40160,40165,40163,40166:GOTO40161
40162
AN$=AN$+A$:IFLEN(AN$)>A1%THENAN$=LEFT$(AN$,A1%):GOTO40161ELSE
PRINT@PO%+A1%-LEN(AN$),AN$;:GOTO40161
40163
S%=-S%:PRINT@PO%+A1%,"";:IFS%=-1THENPRINT"-";ELSEPRINT" ";
40164 GOTO40161
40165 IFINSTR(AN$,".")=0THEN40162ELSE40161
40166 IFAN$=""THEN40168ELSEPRINT@PO%,STRING$(A1%-LEN(AN$),"
");
40167 IFS%=-1THENAN$="-"+AN$:GOTO40169ELSE40169
40168 IF A$=CHR$(91)THENPRINT@PO%,STRING$(A1%,132);"
";ELSEPRINT@PO%,STRING$(A1%," ");" ";
40169 GOSUB 40170:RETURN
40170 AN#=VAL(AN$):AN#=(INT((AN#+.005)*100)/100):RETURN
50000 REM * Data Entry Subroutines                    *
50001 REM * Gosub 40140 - Dollar entry                *
50002 REM * Gosub 40150 - Numeric entry               *
```

```
50003 REM * Gosub 40130 - Alpha/numeric entry                    *
50004 REM * Parameters:                                          *
50005 REM *  Input:  PO% = Print@ position                       *
50006 REM *          A1% = Number of characters to allow        *
50007 REM *  Output: AN$ = Response                              *
50008 REM *          AN# = Rounded numeric equivalent           *
50999 REM * String Compression Subroutine                       *
51000 IF LEN(AN$)=0 THEN RETURN ELSE IF MID$(AN$,1,1)=" "
THEN AN$=RIGHT$(AN$,LEN(AN$)-1):GOTO 51000
51010 IF LEN(AN$)=0 THEN RETURN ELSE IF
MID$(AN$,LEN(AN$),1)=" " THEN AN$=LEFT$(AN$,LEN(AN$)-1):GOTO
51010
51020 RETURN
```

Program Listing: CGMEBF/BAS

```
0 REM * cgmebf/bas version 1 08/09/82
10 CLS:CLEAR10000:FI$="$$#,###,###.##":PRINT">> Option: Post
Or Print Out Record Of Capital Gains For <<":PRINT ">>
Machinery, Equipment, Buildings, Or Farm                    <<"
20 PRINT:PRINT"Do you wish to:":PRINT:PRINT"1) Post an entry
to the record":PRINT"2) Print out the stored
record":PRINT"3) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1, 2, or 3)"
30 PO%=607:A1%=1:GOSUB 40160:A$=AN$:PRINT:IF A$="" OR
(A$<>"1" AND A$<>"2" AND A$<>"3") GOTO 30
40 IF A$="3" THEN CLS:PRINT"Returning to MASTER MENU":RUN
"MASTER/BAS"
50 IF A$="2" GOTO 380
60 CLS:PRINT"Place the disk marked SPECIAL in Drive 1, then
Press <ENTER>"
70 PO%=64:A1%=0:GOSUB40130
80 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:CLOSE:IF
D$="SPECIAL" GOTO 100 ELSE:CLS:PRINT"ERROR - this is not a
SPECIAL disk!  It is marked: ";D$:GOTO 2000
100 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:DIM ST(14):FOR Z=1
TO 14:INPUT#1,ST(Z):NEXT Z:CLOSE
110 CLS:PRINT"Entry #";ST(7)+1
120 PRINT:PRINT"Enter a description of the item (up to 25
chars.)":PO%=192:A1%=25:GOSUB40130:DE$=AN$:PRINT
130 PRINT"Enter the date this item was sold or junked (up to
```

```
8 chars.)":PO%=320:A1%=8:GOSUB40130:DA$=AN$:PRINT
140 PRINT:PRINT"Net Sale Price
":PO%=463:A1%=10:GOSUB40140:NS#=AN#:PRINT
150 PRINT"DePreciation taken in Previous
years":PO%=549:A1%=10:GOSUB40140:DE#=AN#:PRINT
160 PRINT:PRINT"Original
Cost":PO%=654:A1%=10:GOSUB40140:OC#=AN#:PRINT
170 CG#=NS#-(OC#-DE#)
180 CLS:PRINT"Verify that what you typed is correct:":PRINT
190 PRINT"Description: ";DE$
200 PRINT"Date sold or Junked: ";DE$
210 PRINT:PRINT"Net Sale Price: ";:PRINT USING FI$;NS#
220 PRINT"DePreciation taken in Prior years: ";:PRINT USING
FI$;DE#
230 PRINT"Original Cost: ";:PRINT USING FI$;OC#
240 PRINT@960,"Is this information: (1=correct  or
2=incorrect)";
250 PO%=1010:A1%=1:GOSUB40160:A$=AN$:IF A$<>"1" AND A$<>"2"
GOTO 250 ELSE IF A$="2" THEN RUN
260 CLS:PRINT"Do you wish to:":PRINT:PRINT"1) Post this
entry to disk":PRINT"2) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1 or 2)"
270 PO%=347:A1%=1:GOSUB40160:A$=AN$:IF A$<>"1" AND A$<>"2"
GOTO 270 ELSE IF A$="2" THEN A$="3":GOTO 40
280 CLS:PRINT"Place the disk marked SPECIAL in Drive 1, then
Press <ENTER>"
290 PO%=64:A1%=0:GOSUB40130
```

```
300 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:CLOSE:IF
D$="SPECIAL" GOTO 350 ELSE PRINT"ERROR - this is not a
SPECIAL disk!  It is marked: ";D$:PRINT
310 PRINT"Do you wish to:":PRINT:"1) Try a9ain":PRINT"2)
Abort and return to MASTER MENU":PRINT:PRINT"Enter your
choice (1 or 2)"
320 PO%=475:A1%=1:GOSUB40160:A$=AN$:IF A$<>"1" AND A$<>"2"
GOTO 320 ELSE IF A$="2" THEN A$="3":GOTO 40
330 GOTO 280
350 CLS:PRINT"Writing entry to disk...Please stand by":OPEN
"E",1,"c9mebf/dat:1":PRINT#1,DE$;",";DA$;",";NS#;DE#;DC#;CG#:
CLOSE
360 ST(7)=ST(7)+1:OPEN "O",1,"HEADER/DAT":PRINT#1,D$:FOR Z=1
TO 14:PRINT#1,ST(Z):NEXT Z:CLOSE
370 RUN
380 CLS:PRINT"Place the disk marked SPECIAL in Drive 1, then
Press <ENTER>"
390 PO%=64:A1%=0:GOSUB40130
400 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:CLOSE:IF
D$="SPECIAL" GOTO 450 ELSE CLS:PRINT"ERROR - this is not a
SPECIAL disk!  It is marked: ";D$:PRINT
410 PRINT"Do you wish to:":PRINT:PRINT"1) Try
a9ain":PRINT"2) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1 or 2)"
420 PO%=411:A1%=1:GOSUB40160:A$=AN$:IF A$<>"1" AND A$<>"2"
GOTO 420 ELSE IF A$="2"THEN A$="3":GOTO 40
430 GOTO 380
```

```
450 DIM ST(14):OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:FOR Z=1
TO 14:INPUT#1,ST(Z):NEXT Z:CLOSE:IF ST(7)<>0 GOTO 500
460 CLS:PRINT"ERROR - there are no such entries on this
SPECIAL disk!":PRINT:GOTO 410
500 CLS:PRINT"Ready Printer, then Press <ENTER>"
510 POK=128:A1%=0:GOSUB40130
520 A$="CAPITAL GAINS ON MACHINERY, EQUIPMENT, BUILDINGS, &
FARM":LPRINT TAB(33-(LEN(A$)/2));A$:LPRINT
TAB(33-(LEN(A$)/2));STRING$(LEN(A$),"=")
525 OPEN "I",1,"CGMEBF/dat:1":TC#=0
530 LPRINT:LPRINT:FOR Z=1 TO ST(7):LPRINT
STRING$(66,"-"):LPRINT "Entry #";Z:LPRINT
540 INPUT#1,DE$,DA$,NS#,DE#,OC#,CG#
550 LPRINT "Description: ";DE$:LPRINT "Date Sold Or Junked:
";DA$
560 LPRINT:LPRINT"Net Sale Price:";TAB(40);:LPRINT USING
FI$;NS#
570 LPRINT "Depreciation taken in Previous
years:";TAB(40);:LPRINT USING FI$;DE#
580 LPRINT:LPRINT "Original Cost:";TAB(40);:LPRINT USING
FI$;OC#
590 LPRINT "Net Capital Gain Or Loss:";TAB(40);:LPRINT USING
FI$;CG#
600 TC#=TC#+CG#:NEXT Z
610 LPRINT STRING$(66,"-"):LPRINT STRING$(66,"-"):LPRINT
"Total Capital Gain Or Loss:";TAB(40);:LPRINT USING
FI$;TC#:LPRINT STRING$(66,"-"):CLOSE:RUN
```

```
2000 PRINT@960,"Press <ENTER> to
continue";:PO%=1020:A1%=0:GOSUB 40130:RUN
40070 AN$=" ":POKE VARPTR(AN$),A1%:POKE
VARPTR(AN$)+2,INT(PO%/256)+60:POKE
VARPTR(AN$)+1,PO%-INT(PO%/256)*256:RETURN
40130 QS%=PO%:Q9%=A1%:A%=0:PRINT@PO%,STRING$(A1%,132);
40131 IF A%=A1%THEN40134ELSEPRINT@PO%+A%,CHR$(132);
40132 A$=INKEY$:IFA$=""THEN40132ELSEIFINSTR("
!#$%&'()*+<-.?0123456789:;<=>?@/ABCDEFGHIJKLMNOPQRSTUVWXYZabc
defghijklmnopqrstuvwxyz",A$)THENPRINT@PO%+A%,A$;:A%=A%+1:GOTO
40131
40133
ONINSTR(CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40135,40130,40138:GO
TO40131
40134 A$=INKEY$:IF A$=""THEN40134ELSE40133
40135 IFA%<A1%THENPRINT@PO%+A%,CHR$(132);
40136 A%=A%-1:IFA%<0THENA%=0:GOTO40131ELSE40131
40137 A%=0
40138
IFA$=CHR$(31)THENPRINT@PO%,STRING$(A1%,132);ELSEPRINT@PO%+A%,
STRING$(A1%-A%," ");
40139 GOSUB40070:GOTO 51000
40140 S%=1:AN$="":PRINT@PO%,"$";STRING$(A1%,132);"
";:PRINT@PO%+A1%-2,".";
40141
A$=INKEY$:IFA$=""THEN40141ELSEIFINSTR("0123456789",A$)THEN401
43ELSEONINSTR("-"+CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40145,4014
```

```
0,40140,40147
40142 GOTO40141
40143
AN$=AN$+A$:IFLEN(AN$)=1THENAN$=CHR$(132)+AN$ELSEIFLEN(AN$)>A1
%-1THENAN$=LEFT$(AN$,A1%-1)ELSEIFLEN(AN$)=3ANDLEFT$(AN$,1)=CH
R$(132)THENAN$=RIGHT$(AN$,2)
40144
PRINT@PO%+A1%-LEN(AN$),LEFT$(AN$,LEN(AN$)-2);".";RIGHT$(AN$,2
);:GOTO40141
40145
S%=-S%:PRINT@PO%+A1%+1,"";:IFS%=-1THENPRINT"-";:GOTO40141ELSE
PRINT" ";:GOTO40141
40146 IF A$=CHR$(91)THENPRINT@PO%+1,STRING$(A1%,132);"
";:PRINT@PO%+A1%-2,".";:GOTO40149ELSEPRINT@PO%,STRING$(A1%+2,
" ");:GOTO40149
40147 IFLEN(AN$)=0THENPRINT@PO%,STRING$(A1%+2,"
");:GOTO40149 ELSEPRINT@PO%+1,STRING$(A1%-1-LEN(AN$),"
");:IFLEFT$(AN$,1)=CHR$(132)THENPRINT@PO%+A1%-1,"0";:MID$(AN$
,1,1)="0"
40148
AN$=MID$(AN$,1,LEN(AN$)-2)+"."+RIGHT$(AN$,2):IFS%=-1THENAN$="
-"+AN$
40149 GOSUB 40170:RETURN
40160 S%=1:AN$="":PRINT@PO%,STRING$(A1%,132);" ";
40161
A$=INKEY$:IFA$=""THEN40161ELSEIFINSTR("0123456789",A$)THEN401
62ELSEONINSTR(CHR$(8)+CHR$(31)+".+"-"+CHR$(13),A$)GOTO40160,
```

```
40160,40165,40163,40166:GOTO40161
40162
AN$=AN$+A$:IFLEN(AN$)>A1%THENAN$=LEFT$(AN$,A1%):GOTO40161ELSE
PRINT@PO%+A1%-LEN(AN$),AN$;:GOTO40161
40163
S%=-S%:PRINT@PO%+A1%,"";:IFS%=-1THENPRINT"-";ELSEPRINT" ";
40164 GOTO40161
40165 IFINSTR(AN$,".")=0THEN40162ELSE40161
40166 IFAN$=""THEN40168ELSEPRINT@PO%,STRING$(A1%-LEN(AN$),"
");
40167 IFS%=-1THENAN$="-"+AN$:GOTO40169ELSE40169
40168 IF A$=CHR$(91)THENPRINT@PO%,STRING$(A1%,132);"
";ELSEPRINT@PO%,STRING$(A1%," ");" ";
40169 GOSUB 40170:RETURN
40170 AN#=VAL(AN$):AN#=(INT((AN#+.005)*100)/100):RETURN
50999 REM * String Compression subroutine
*
51000 IF LEN(AN$)=0 THEN 51020 ELSE IF MID$(AN$,1,1)=" "
THEN AN$=RIGHT$(AN$,LEN(AN$)-1):GOTO 51000
51010 IF LEN(AN$)=0 THEN 51020 ELSE IF
MID$(AN$,LEN(AN$),1)=" " THEN AN$=LEFT$(AN$,LEN(AN$)-1):GOTO
51010
51020 IF (LEN(AN$)=0 AND Q9%<>0) THEN PO%=Q8%:A1%=Q9%:GOTO
40130
51030 RETURN
```

Program Listing: CGPBLSD/BAS

```
 0 REM * cgpblsd/bas version 1 08/09/82
 10 CLS:CLEAR 10000:FI$="$$#,###,###.##":PRINT"<< Option:
 Post or Print Out Record of Capital Gains For <<":PRINT">>
 Purchased Breeding Livestock Sold or Died              <<"
 20 PRINT:PRINT"Do you wish to:":PRINT:PRINT"1) Post entry to
 capital gains record":PRINT"2) Print out a stored
 record":PRINT"3) Abort and return to MASTER
 MENU":PRINT:PRINT"Enter your choice (1, 2, OR 3)"
 30 PO%=607:A1%=1:GOSUB40160:A$=AN$:IF A$="" OR (A$<>"1" AND
 A$<>"2" AND A$<>"3") GOTO 30
 40 IF A$="3" THEN CLS:PRINT"Returning to MASTER MENU":RUN
 "MASTER/BAS"
 50 IF A$="2" GOTO 320
 60 CLS:PRINT"Place the disk marked SPECIAL in Drive 1 then
 Press <ENTER>"
 70 PO%=128:A1%=0:GOSUB40130
 80 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:CLOSE:IF
 D$<>"SPECIAL" THEN CLS:PRINT"ERROR - this is not a SPECIAL
 disk!  It is marked: ";D$:GOTO 2000
 90 DIM ST(14):OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:FOR Z=1
 TO 14:INPUT#1,ST(Z):NEXT:CLOSE
 100 CLS:PRINT"Entry #";ST(6)+1:PRINT
 110 PRINT"Enter date sold or died (up to 8
 chars.)":PO%=169:A1%=8:GOSUB40130:DA$=AN$:PRINT
 120 PRINT"Enter number of animals sold or
```

```
died":PO%=229:A1%=5:GOSUB40160:NA=VAL(AN$):PRINT
130 PRINT:PRINT"Net Sale
Price":PO%=335:A1%=10:GOSUB40140:SV#=AN#:PRINT
140 PRINT"Unrecovered
Cost":PO%=401:A1%=10:GOSUB40140:UC#=AN#:PRINT
150 PRINT:PRINT"Capital Gain or
Loss":PO%=533:A1%=10:GOSUB40140:CG#=AN#:PRINT
160 PRINT"Investment Tax Credit
Recaptured":PO%=609:A1%=10:GOSUB40140:IC#=AN#:PRINT
170 CLS:PRINT"Verify that what you typed is correct:":PRINT
180 PRINT:PRINT"Date sold or died: ";DA$:PRINT"Number of
animals sold or died:";NA
190 PRINT:PRINT"Net Sale Price:";:PRINT USING FI$;SV#
200 PRINT"Unrecovered Cost:";:PRINTUSING FI$;UC#
210 PRINT:PRINT"Capital Gain or Loss:";:PRINT USING FI$;CG#
220 PRINT"Investment Tax Credit Recaptured:";:PRINT USING
FI$;IC#
230 PRINT@960,"Is this information (1=correct  or
2=incorrect)";:PO%=1009:A1%=1:GOSUB40160:A$=AN$:IF A$="" OR
(A$<>"1" AND A$<>"2") GOTO 230 ELSE IF A$="2" THEN
A$="3":GOTO 40
240 CLS:PRINT"Do you wish to:":PRINT:PRINT"1) Post this
entry to disk":PRINT"2) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1 or 2)"
250 PO%=347:A1%=1:GOSUB40160:A$=AN$:IF A$="" OR (A$<>"1" AND
A$<>"2") GOTO 250 ELSE IF A$="2" THEN A$="3":GOTO 40
260 CLS:PRINT"Place the disk marked SPECIAL in Drive 1, then
```

```
Press <ENTER>":PO%=128:A1%=0:GOSUB40130
270 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:CLOSE:IF
D$="SPECIAL" GOTO 300 ELSE CLS:PRINT"ERROR - this is not a
SPECIAL disk!  It is marked: ";D$
280 PRINT:PRINT"Do you wish to:":PRINT:PRINT"1) Try
again":PRINT"2) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1 or 2)"
285 PO%=475:A1%=1:GOSUB 40160:A$=AN$:IF A$="" OR (A$<>"1"
AND A$<>"2") GOTO 285 ELSE IF A$="2" THEN A$="3":GOTO 40
290 GOTO 260
300 CLS:PRINT"Writing entry to disk - Please stand by":OPEN
"E",1,"CGPBLSD/DAT:1":PRINT#1,DA$;",";NA;SV#;UC#;CG#;IC#:CLOS
E
310 ST(6)=ST(6)+1:OPEN "O",1,"HEADER/DAT":PRINT#1,D$:FOR Z=1
TO 14:PRINT#1,ST(Z):NEXTZ:CLOSE:RUN
320 CLS:PRINT"Place the disk marked SPECIAL in Drive 1, then
Press <ENTER>":PO%=128:A1%=0:GOSUB40130
330 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:CLOSE:IF
D$="SPECIAL" GOTO 400 ELSE CLS:PRINT"ERROR - this is not a
SPECIAL disk!  It is marked: ";D$
340 PRINT:PRINT"Do you wish to:":PRINT:PRINT"1) Try
again":PRINT"2) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1 or 2)"
350 PO%=475:A1%=1:GOSUB40160:A$=AN$:IF A$="" OR (A$<>"1" AND
A$<>"2") GOTO 350 ELSE IF A$="2" THEN A$="3":GOTO 40
360 GOTO 320
400 DIM ST(14):OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:FOR Z=1
```

```
TO 14:INPUT#1,ST(Z):NEXT Z:CLOSE:IF ST(6)<>0 GOTO 410 ELSE
CLS:PRINT"ERROR - There are no such entries posted on this
SPECIAL disk!":GOTO 340
410 CLS:PRINT"Ready Printer, then Press <ENTER>"
420 PO%=64:A1%=0:GOSUB40130
430 A$="CAPITAL GAINS FOR PURCHASED BREEDING LIVESTOCK SOLD
OR DIED":LPRINT TAB(33-(LEN(A$)/2));A$:LPRINT
TAB(33-(LEN(A$)/2));STRING$(LEN(A$),"="):LPRINT:LPRINT
435 TC#=0
440 OPEN "i",1,"CGPBLSD/DAT":FOR Z=1 TO ST(6):LPRINT
STRING$(66,"-"):INPUT#1,DA$,NA,SV#,UC#,CG#,IC#
450 TC#=TC#+CG#
460 LPRINT "Entry #";Z:LPRINT
470 LPRINT "Date sold or died:";TAB(40);DA$
480 LPRINT "Number of animals sold or died:";TAB(40);NA
490 LPRINT:LPRINT"Net Sale Price:";TAB(40);:LPRINTUSING
FI$;SV#
500 LPRINT"Unrecovered Cost:";TAB(40);:LPRINTUSING FI$;UC#
510 LPRINT:LPRINT"Capital Gain or Loss:";TAB(40);:LPRINT
USING FI$;CG#
520 LPRINT"Investment Tax Credit
Recaptured:";TAB(40);:LPRINT USING FI$;IC#
530 NEXT Z
540 LPRINT STRING$(66,"-"):LPRINT STRING$(66,"-"):LPRINT
"Total Capital Gain or Loss:";TAB(40);:LPRINT USING
FI$;TC#:LPRINT STRING$(66,"-"):CLOSE:RUN
2000 PRINT@960,"Press <ENTER> to
```

```
continue";:PO%=1020:A1%=0:GOSUB40130:RUN
40070 AN$=" ":POKE VARPTR(AN$),A1%:POKE
VARPTR(AN$)+2,INT(PO%/256)+60:POKE
VARPTR(AN$)+1,PO%-INT(PO%/256)*256:RETURN
40130 Q8%=PO%:Q9%=A1%:A%=0:PRINT@PO%,STRING$(A1%,132);
40131 IF A%=A1%THEN40134ELSEPRINT@PO%+A%,CHR$(132);
40132 A$=INKEY$:IFA$=""THEN40132ELSEIFINSTR("
!#$%&'()*+<-.?0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ/abc
defghijklmnoPqrstuvwxyz",A$)THENPRINT@PO%+A%,A$;:A%=A%+1:GOTO
40131
40133
ONINSTR(CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40135,40130,40138:GO
TO40131
40134 A$=INKEY$:IF A$=""THEN40134ELSE40133
40135 IFA%<A1%THENPRINT@PO%+A%,CHR$(132);
40136 A%=A%-1:IFA%<0THENA%=0:GOTO40131ELSE40131
40137 A%=0
40138
IFA$=CHR$(91)THENPRINT@PO%,STRING$(A1%,132);ELSEPRINT@PO%+A%,
STRING$(A1%-A%," ");
40139 GOSUB40070:GOTO51000
40140 S%=1:AN$="":PRINT@PO%,"$";STRING$(A1%,132);"
";:PRINT@PO%+A1%-2,".";
40141
A$=INKEY$:IFA$=""THEN40141ELSEIFINSTR("0123456789",A$)THEN401
43ELSEONINSTR("-"+CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40145,4014
0,40140,40147
```

```
40142 GOTO40141
40143
AN$=AN$+A$:IFLEN(AN$)=1THENAN$=CHR$(132)+AN$ELSEIFLEN(AN$)>A1
%-1THENAN$=LEFT$(AN$,A1%-1)ELSEIFLEN(AN$)=3ANDLEFT$(AN$,1)=CH
R$(132)THENAN$=RIGHT$(AN$,2)
40144
PRINT@PO%+A1%-LEN(AN$),LEFT$(AN$,LEN(AN$)-2);".";RIGHT$(AN$,2
):GOTO40141
40145
S%=-S%:PRINT@PO%+A1%+1,"";:IFS%=-1THENPRINT"-";:GOTO40141ELSE
PRINT" ";:GOTO40141
40146 IF A$=CHR$(91)THENPRINT@PO%+1,STRING$(A1%,132);"
":PRINT@PO%+A1%-2,".":GOTO40149ELSEPRINT@PO%,STRING$(A1%+2,
" ");:GOTO40149
40147 IFLEN(AN$)=0THENPRINT@PO%,STRING$(A1%+2,"
");:GOTO40149 ELSEPRINT@PO%+1,STRING$(A1%-1-LEN(AN$),"
");:IFLEFT$(AN$,1)=CHR$(132)THENPRINT@PO%+A1%-1,"0";:MID$(AN$
,1,1)="0"
40148
AN$=MID$(AN$,1,LEN(AN$)-2)+"."+RIGHT$(AN$,2):IFS%=-1THENAN$="
-"+AN$
40149 GOSUB 40170:RETURN
40160 S%=1:AN$="":PRINT@PO%,STRING$(A1%,132);" ";
40161
A$=INKEY$:IFA$=""THEN40161ELSEIFINSTR("0123456789",A$)THEN401
52ELSEONINSTR(CHR$(8)+CHR$(31)+"."+"-"+CHR$(13),A$)GOTO40160,
40160,40165,40163,40166:GOTO40161
```

```
40162
AN$=AN$+A$:IFLEN(AN$)>A1%THENAN$=LEFT$(AN$,A1%):GOTO40161ELSE
PRINT@PO%+A1%-LEN(AN$),AN$;:GOTO40161
40163
S%=-S%:PRINT@PO%+A1%,"";:IFS%=-1THENPRINT"-";ELSEPRINT" ";
40164 GOTO40161
40165 IFINSTR(AN$,".")=0THEN40162ELSE40161
40166 IFAN$=""THEN40168ELSEPRINT@PO%,STRING$(A1%-LEN(AN$),"
");
40167 IFS%=-1THENAN$="-"+AN$:GOTO40169ELSE40169
40168 IF A$=CHR$(91)THENPRINT@PO%,STRING$(A1%,132);"
";ELSEPRINT@PO%,STRING$(A1%," ");" ";
40169 GOSUB 40170:RETURN
40170 AN#=VAL(AN$):AN#=(INT((AN#+.005)*100)/100):RETURN
50000 REM * Data Entry Subroutines                          *
50001 REM * Gosub 40140 - Dollar entry                      *
50002 REM * Gosub 40150 - Numeric entry                     *
50003 REM * Gosub 40130 - Alpha/numeric entry               *
50004 REM * Parameters:                                     *
50005 REM *  input:  po% = print @ position                 *
50006 REM *          a1% = number of characters to allow   *
50007 REM * output:  an$ = response                         *
50008 REM *          an# = numeric rounded equivalent       *
50999 REM * String Compression subroutine
*
51000 IF LEN(AN$)=0 THEN 51020 ELSE IF MID$(AN$,1,1)=" "
THEN AN$=RIGHT$(AN$,LEN(AN$)-1):GOTO 51000
```

```
51010 IF LEN(AN$)=0 THEN 51020 ELSE IF
MID$(AN$,LEN(AN$),1)=" " THEN AN$=LEFT$(AN$,LEN(AN$)-1):GOTO
51010
51020 IF (LEN(AN$)=0 AND Q9%<>0) THEN P0%=Q8%:A1%=A9%:GOTO
40130
```

Program Listing: CGRBL/BAS

```
0 REM * cgrbl/bas version 2 08/05/82
10 CLS:CLEAR 10000:DIM ST(14):FI$="$$#,###,###.##":PRINT">>
Option: Post or Print Out Record of Capital Gains For
<<":PRINT">> Raised Breeding Livestock Sold
     <<"
15 FB$="############"
20 PRINT:PRINT"Do you wish to:":PRINT:PRINT"1) Post an entry
to capital gain record":PRINT"2) Print out capital gain
record":PRINT:PRINT"3) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1, 2, or 3)"
30 PO%=671:A1%=1:GOSUB40160:A$=AN$:PRINT:IF A$="" OR
(A$<>"1" AND A$<>"2" AND A$<>"3") GOTO 30
40 IF A$="3" THEN CLS:PRINT"Returning to MASTER MENU":RUN
"MASTER/BAS"
50 IF A$="2" GOTO 340
60 CLS:PRINT"Place the disk marked SPECIAL in Drive
1,":PRINT"then Press <ENTER>."
70 PO%=128:A1%=0:GOSUB40130
80 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:CLOSE
90 IF D$="SPECIAL" GOTO 110 ELSE CLS:PRINT"ERROR - this is
not a SPECIAL disk!  It is marked: ";D$
100 GOTO 2000
110 OPEN "i",1,"HEADER/DAT:1":INPUT#1,D$:FOR Z=1 TO
14:INPUT#1,ST(Z):NEXT Z:CLOSE
120 EN=ST(5):CLS:PRINT"There are";EN;"entries in record."
```

```
130 EN=EN+1
140 PRINT"Entry #";EN
150 DA$="":PRINT "Date sold (up to 8 chars.)
":PO%=155:A1%=8:GOSUB40130:DA$=AN$:PRINT
160 IF LEN(DA$)<>0 GOTO 195
165 CLS:PRINT"ERROR - you must type in a date!":GOTO 2200
170 CLS:PRINT"Do you wish to:":PRINT:PRINT"1) Retype last
entry":PRINT"2) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1 or 2)"
180 PO%=347:A1%=1:GOSUB40160:A$=AN$:PRINT:IF A$="" OR
(A$<>"1" AND A$<>"2") GOTO 180 ELSE IF A$="2" THEN
A$="3":GOTO 40
190 CLS:PRINT:GOTO 140
195 IF LEN(DA$)>8 THEN DA$=LEFT$(DA$,8)
200 PRINT:PRINT "Number of animals sold
":PO%=279:A1%=5:GOSUB40160:NS=VAL(AN$):PRINT
205 PRINT
210 PRINT "Total amount ($) received ":PO%=411:A1%=10:GOSUB
40140:TR#=AN#:PRINT
220 CLS:PRINT"Verify that what you typed is
correct:":PRINT:PRINT"Entry #";EN
230 PRINT"Date sold: ";DA$:PRINT:PRINT"Number of animals
sold: ";NS:PRINT:PRINT"Total amount received: ";:PRINT USING
FI$;TR#:PRINT:PRINT@960,"Is this information: (1=correct  or
 2=incorrect)";
240 PO%=1010:A1%=1:GOSUB40160:A$=AN$:IF A$="" OR (A$<>"1"
AND A$<>"2") GOTO 240 ELSE IF A$="2" CLS:GOTO 2200
```

```
250 CLS:PRINT"Mount the disk marked SPECIAL in DRIVE
1,":PRINT"Then Press <ENTER>."
260 PO%=128:A1%=0:GOSUB40130:PRINT
270 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:CLOSE
280 IF D$="SPECIAL" GOTO 310
290 CLS:PRINT"ERROR - this is not a SPECIAL disk!  It is
marked: ";D$:PRINT:PRINT"Do you wish to:":PRINT:PRINT"1) Try
again":PRINT"2) Abort and return to MASTER
MENU":PRINT:PRINT"Enter your choice (1 or 2)
300 PO%=475:A1%=1:GOSUB40160:A$=AN$:PRINT:IF A$="" OR
(A$<>"1" AND A$<>"2") GOTO 300 ELSE IF A$="2" THEN
A$="3":GOTO 40
305 GOTO 250
310 CLS:PRINT"Writing entry #";EN;"to disk - Please stand
by.":OPEN "E",1,"CGRBL/DAT:1":PRINT#1,DA$;",";NS;TR#:CLOSE
320 OPEN "O",1,"HEADER/DAT:1":ST(5)=EN:PRINT#1,D$;",";:FOR
Z=1 TO 14:PRINT#1,ST(Z);:NEXT Z:CLOSE
330 RUN
340 CLS:PRINT"Place the disk marked SPECIAL in Drive
1,":PRINT"then Press any key."
350 PO%=128:A1%=0:GOSUB40130:PRINT
360 OPEN "I",1,"header/dat:1":INPUT#1,D$:CLOSE
370 IF D$="SPECIAL" GOTO 400 ELSECLS:PRINT"ERROR - this is
not a SPECIAL disk!  It is marked: ";D$:PRINT:PRINT"Do you
wish to:":PRINT:PRINT"1) Try again":PRINT"2) Abort and
return to MASTER MENU":PRINT:PRINT"Enter your choice (1 or
2)"
```

```
380 PO%=475:A1%=1:GOSUB40160:A$=AN$:PRINT:IF A$="" OR
(A$<>"1" AND A$<>"2") GOTO 380 ELSE IF A$="2" THEN
A$="3":GOTO 40
390 GOTO 340
400 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:FOR Z=1 TO
14:INPUT#1,ST(Z):NEXT Z:CLOSE
410 IF ST(5)<>0 GOTO 450 ELSECLSPRINT"ERROR - No entries
have yet been stored on this disk!":PRINT:PRINT"Do you wish
to":PRINT:PRINT"1) Try again":PRINT"2) Abort and return to
MASTER MENU":PRINT:PRINT"Enter your choice (1 or 2)"
420 PO%=475:A1%=1:GOSUB40160:A$=AN$:PRINT:IF A$="" OR
(A$<>"1" AND A$<>"2") GOTO 420 ELSE IF A$="2" THEN
A$="3":GOTO 40
430 GOTO 340
450 CLS:PRINT"Ready Printer, then press any key"
460 PO%=128:A1%=0:GOSUB40130:PRINT
470 A$="CAPITAL GAINS FOR RAISED BREEDING LIVESTOCK
SOLD":LPRINT TAB(33-(LEN(A$)/2));A$:LPRINT
TAB(33-(LEN(A$)/2));STRING$(LEN(A$),"=")
475 LPRINT:OPEN "I",1,"CGRBL/DAT":TG#=0
480 FOR Z=1 TO ST(5):LPRINT STRING$(66,"-"):LPRINT "Entry
#";Z;"of";ST(5):LPRINT
490 INPUT#1,DA$,NS,TR#:LPRINT "Date sold:";TAB(56);DA$
500 LPRINT:LPRINT "Number of animals sold:";TAB(52);:LPRINT
USING FB$;NS
510 LPRINT:LPRINT "Total amount received:";TAB(50);:LPRINT
USING FI$;TR#
```

```
520 LPRINT:TG#=TG#+TR#:NEXT Z
530 LPRINT STRING$(66,"-"):LPRINT STRING$(66,"-")
540 LPRINT "Total capital gains:";TAB(50);:LPRINT USING
F1$;TG#
550 LPRINT STRING$(66,"-"):RUN
2000 PRINT@960,"Press <ENTER> to
continue";:PO%=1020:A1%=0:GOSUB40130:CLS:RUN
2200 PRINT@960,"Press <ENTER> to
continue";:PO%=1020:A1%=0:GOSUB40130:GOTO 170
40070 AN$=" ":POKE VARPTR(AN$),A1%:POKE
VARPTR(AN$)+2,INT(PO%/256)+60:POKE
VARPTR(AN$)+1,PO%-INT(PO%/256)*256:RETURN
40130 A%=0:PRINT@PO%,STRING$(A1%,132);
40131 IF A%=A1%THEN40134ELSEPRINT@PO%+A%,CHR$(132);
40132 A$=INKEY$:IFA$=""THEN40132ELSEIFINSTR("
!#$%&'()*+<-.?0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ^abc
def9hijklmnoP9rstuvwxyz",A$)THENPRINT@PO%+A%,A$;:A%=A%+1:GOTO
40131
40133
ONINSTR(CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40135,40130,40138:GO
TO40131
40134 A$=INKEY$:IF A$=""THEN40134ELSE40133
40135 IFA%<A1%THENPRINT@PO%+A%,CHR$(132);
40136 A%=A%-1:IFA%<0THENA%=0:GOTO40131ELSE40131
40137 A%=0
40138
IFA$=CHR$(91)THENPRINT@PO%,STRING$(A1%,132);ELSEPRINT@PO%+A%,
```

```
STRING$(A1%-A%," ");
40139 GOSUB40070:GOTO51000
40140 S%=1:AN$="":PRINT@PO%,"$";STRING$(A1%,132);"
";:PRINT@PO%+A1%-2,".";
40141
A$=INKEY$:IFA$=""THEN40141ELSEIFINSTR("0123456789",A$)THEN401
43ELSEONINSTR("-"+CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40145,4014
0,40140,40147
40142 GOTO40141
40143
AN$=AN$+A$:IFLEN(AN$)=1THENAN$=CHR$(132)+AN$ELSEIFLEN(AN$)>A1
%-1THENAN$=LEFT$(AN$,A1%-1)ELSEIFLEN(AN$)=3ANDLEFT$(AN$,1)=CH
R$(132)THENAN$=RIGHT$(AN$,2)
40144
PRINT@PO%+A1%-LEN(AN$),LEFT$(AN$,LEN(AN$)-2);".";RIGHT$(AN$,2
);:GOTO40141
40145
S%=-S%:PRINT@PO%+A1%+1,"";:IFS%=-1THENPRINT"-";:GOTO40141ELSE
PRINT" ";:GOTO40141
40146 IF A$=CHR$(91)THENPRINT@PO%+1,STRING$(A1%,132);"
";:PRINT@PO%+A1%-2,".";:GOTO40149ELSEPRINT@PO%,STRING$(A1%+2,
" ");:GOTO40149
40147 IFLEN(AN$)=0THENPRINT@PO%,STRING$(A1%+2,"
");:GOTO40149 ELSEPRINT@PO%+1,STRING$(A1%-1-LEN(AN$),"
");:IFLEFT$(AN$,1)=CHR$(132)THENPRINT@PO%+A1%-1,"0";:MID$(AN$
,1,1)="0"
40148
```

```
AN$=MID$(AN$,1,LEN(AN$)-2)+","+RIGHT$(AN$,2):IFS%=-1THENAN$="
-"+AN$
40149 GOSUB 40170:RETURN
40160 S%=1:AN$="":PRINT@PO%,STRING$(A1%,132);" ";
40161
A$=INKEY$:IFA$=""THEN40161ELSEIFINSTR("0123456789",A$)THEN401
62ELSEONINSTR(CHR$(8)+CHR$(31)+"."+"-"+CHR$(13),A$)GOTO40160,
40160,40165,40163,40166:GOTO40161
40162
AN$=AN$+A$:IFLEN(AN$)>A1%THENAN$=LEFT$(AN$,A1%):GOTO40161ELSE
PRINT@PO%+A1%-LEN(AN$),AN$;:GOTO40161
40163
S%=-S%:PRINT@PO%+A1%,"";:IFS%=-1THENPRINT"-";ELSEPRINT" ";
40164 GOTO40161
40165 IFINSTR(AN$,".")=0THEN40162ELSE40161
40166 IFAN$=""THEN40168ELSEPRINT@PO%,STRING$(A1%-LEN(AN$),"
")
40167 IFS%=-1THENAN$="-"+AN$:GOTO40169ELSE40169
40168 IF A$=CHR$(91)THENPRINT@PO%,STRING$(A1%,132);"
";ELSEPRINT@PO%,STRING$(A1%," ");" ";
40169 GOSUB 40170:RETURN
40170 AN#=VAL(AN$):AN#=(INT((AN#+.005)*100)/100):RETURN
50000 REM * Data Entry Subroutines                    *
50001 REM * Gosub 40140 - Dollar entry                *
50002 REM * Gosub 40150 - Numeric entry               *
50003 REM * Gosub 40130 - Alpha/numeric entry         *
50004 REM * Parameters:                               *
```

```
50005 REM *  inPut: Po% = Print @ Position                    *
50006 REM *          al% = number of characters to allow      *
50007 REM * outPut: an$ = resPonse                            *
50008 REM *          an# = numeric rounded equivalent         *
50999 REM * String ComPression Subroutine                     *
51000 IF LEN(AN$)=0 THEN RETURN ELSE IF MID$(AN$,1,1)=" "
THEN AN$=RIGHT$(AN$,LEN(AN$)-1):GOTO 51000
51010 IF LEN(AN$)=0 THEN RETURN ELSE IF
MID$(AN$,LEN(AN$),1)=" " THEN AN$=LEFT$(AN$,LEN(AN$)-1):GOTO
51010
51020 RETURN
```

Program Listing: CONFIG/BAS

```
0 REM config/bas: revision 2 08/04/82
1 CLEAR 10000
10 CLS:PRINT"System Configuration Editor":PRINT:PRINT"Would
you like to:":PRINT:PRINT"1) Create CONFIG/DAT":PRINT"2)
View or Print out current CONFIG/DAT":PRINT"3) Change
CONFIG/DAT"
11 PRINT "4) Sort CONFIG/DAT by code numbers.":PRINT"5) Sort
CONFIG/DAT by description.":PRINT:PRINT"6) Abort and return
to BASIC":PRINT:PRINT:PRINT"Enter your choice (1, 2, 3, 4,
5, or 6)."
20 PO%=873:A1%=1:GOSUB 40130:A$=AN$:IF A$<>"1" AND A$<>"2"
AND A$<>"3" AND A$<>"4" AND A$<>"5" AND A$<>"6"GOTO20
30 A%=VAL(A$):IF A%=6 THEN CLS:PRINT"System Configuration
Editor finished.":END
50 CLS:PRINT "Data disk is in what drive (0, 1, 2, or 3)"
55 PO%=44:A1%=1:GOSUB 40160:DR$=AN$:IF DR$<>"0" AND DR$<>"1"
AND DR$<>"2" AND DR$<>"3" GOTO 55 ELSE PRINT
57 IF ZZ%=0 AND A%=4 THEN Z9$="numerical code numbers."
60 ON A% GOTO 65,65,65,6000,7000
65 DEFDBL A-Z:ON A% GOTO 70,510,860
70 PRINT"Press <ENTER> when data disk is mounted in Drive
";DR$;:PO%=500:A1%=0:GOSUB 40130
80 OPEN "O",1,"CONFIG/DAT:"+DR$
90
REM"=========================================================="
```

```
100 REM              PREPARING TO WRITE CONFIG/DAT
110
REM"=============================================================="
120 CLS:PRINT "How many account codes ";:PO%=26:A1%=4:GOSUB
40160:XA%=VAL(AN$)
130 PRINT#1,XA%
140 FOR Z%=1 TO XA%:CLS:PRINT "Account code number";Z%;"
of";XA%
145 PRINT
150 PRINT "Enter account code number
";:PO%=156:A1%=5:GOSUB40160:XD=VAL(AN$)
160 PRINT:PRINT "Enter account code - ";XD;"
description":PO%=256:A1%=50:GOSUB40130:XD$=AN$
170 PRINT:PRINT "Enter account type ";:PO%=340:A1%=2:GOSUB
40160:XT=VAL(AN$)
180 PRINT#1,XD;XD$;",";XT
190 NEXT Z%
200 CLS:PRINT "How many bank/account codes
":PO%=29:A1%=4:GOSUB 40160:XB%=VAL(AN$):IF XB%=0 THEN
PRINT#1,0:GOTO 300
210 PRINT#1,XB%:FOR Z%=1 TO XB%:CLS:PRINT "Bank/Account
Code";Z%;" of";XB%
220 PRINT "Enter Bank/Account code ";:PO%=89:A1%=3:GOSUB
40160:XC=VAL(AN$)
230 PRINT:PRINT "Enter description for Bank/Account code
";XC
240 PO%=256:A1%=50:GOSUB 40130:XC$=AN$:PRINT#1,XC;XC$:NEXT
```

```
Z%
300 CLS:PRINT "How many enterPrise codes
":PO%=27:A1%=4:GOSUB 40160:XE%=VAL(AN$):IF XE%=0 THEN
PRINT#1,0:GOTO400
310 PRINT#1,XE%:FOR Z%=1 TO XE%:CLS:PRINT"EnterPrise ";Z%;"
of";XE%
320 PRINT:PRINT"EnterPrise code ";:PO%=145:A1%=5:GOSUB
40160:XF=VAL(AN$)
330 PRINT:PRINT"EnterPrise description":PO%=256:A1%=50:GOSUB
40130:XF$=AN$
340 PRINT#1,XF;XF$:NEXT Z%
400 CLS:PRINT"Number of farm oPerators
":PO%=26:A1%=4:GOSUB40160:XG%=VAL(AN$):IF XG%=0 THEN
PRINT#1,0:GOTO 500
410 PRINT#1,XG%:FORZ%=1 TO XG%:CLS:PRINT"OPerator code";Z%;"
of";XG%:PRINT
420 PRINT "OPerator code
";:PO%=142:A1%=5:GOSUB40160:XH=VAL(AN$)
430 PRINT:PRINT"OPerator
description":PO%=256:A1%=50:GOSUB40130:XH$=AN$
440 PRINT#1,XH;XH$:NEXT Z%
500 CLS:PRINT"File CONFIG/DAT has been created":CLOSE:RUN
510
REM"==========================================="
520 REM"          View current CONFIG/DAT
530
```

```
REM"=============================================================="
540 OPEN "I",1,"CONFIG/DAT:"+DR$
550 CLS:PRINT "Do you wish a hardcopy printout of the file
(Y/N)?"
560 PO%=52:A1%=1:GOSUB40130:A$=AN$:IF A$<>"Y" AND A$<>"y"
AND A$<>"N"AND A$<>"n" GOTO 560 ELSE PRINT
570 IF A$="Y" OR A$="y" THEN LP=1 ELSE LP=0
580 IF LP=1 PRINT:PRINT "Ready Printer, then Press
<ENTER>":PO%=500:A1%=0:GOSUB40130:PRINT
590 A9$="MASTER ACCOUNT CODE CATALOG":IF LP=1 THEN LPRINT
TAB(33-(LEN(A9$)/2));A9$:LPRINTTAB(33-(LEN(A9$)/2));STRING$(L
EN(A9$),"="):LPRINT:LPRINT
600 INPUT#1,XA%:PRINT "Number of account codes = ";XA%
610 IF LP=1 LPRINT "ACCOUNT CODES: Expenditures"
620 IF LP=1 LPRINT "-----------------------------"
630 DIM XD(XA%),XD$(XA%),XT(XA%):FOR Z%=1 TO
XA%:INPUT#1,XD(Z%),XD$(Z%),XT(Z%):NEXT Z%
640 PRINT "ACCOUNT CODES: Expenditures":FOR Z%=1 TO XA%
650 IF XD(Z%)>9999 AND XD(Z%)<20000 THEN PRINT XD(Z%);"
----- ";XD$(Z%):PRINT TAB(10)"Account type - ";XT(Z%):IF
LP=1 THEN LPRINT XD(Z%);" ----- ";XD$(Z%):LPRINT
660 NEXT Z%
665 IF LP=1 THEN CLS:PRINT "Ready Printer for Receipt codes
then Press <ENTER>":PO%=500:A1%=0:GOSUB40130:PRINT
667 IF LP=1 THEN
LPRINTTAB(33-(LEN(A9$)/2));A9$:LPRINTTAB(33-(LEN(A9$)/2));STR
ING$(LEN(A9$),"="):LPRINT:LPRINT
```

```
670 IF LP=1 THEN LPRINT:LPRINT:LPRINT "ACCOUNT CODES:
Receipts":LPRINTSTRING$(23,"-")
680 PRINT "ACCOUNT CODES: Receipts":FOR Z%=1 TO XA%
690 IF XD(Z%)>19999 AND XD(Z%)<30000 THEN PRINT XD(Z%);"
----- ";XD$(Z%):PRINT TAB(10)"Account type - ";XT(Z%):IF
LP=1 THEN LPRINT XD(Z%);" ------ ";XD$(Z%):LPRINT
692 NEXT Z%
695 INPUT#1,XB%:IF XB%=0 PRINT "No Bank/Account
codes.":GOTO740:ELSE PRINT XB%;" Bank/Account codes."
700 IF LP=1 THEN CLS:PRINT"Ready printer for Bank/Account
codes then Press <ENTER>":PO%=500:A1%=0:GOSUB40130:PRINT
710 A$="MASTER BANK/ACCOUNT CODE CATALOG":IF LP=1 THEN
LPRINT TAB(33-(LEN(A$)/2));A$:LPRINT
TAB(33-(LEN(A$)/2));STRING$(LEN(A$),"="):LPRINT:LPRINT
720 FOR Z%=1 TO XB%:INPUT#1,XC,XC$:PRINT XC;" ------ ";XC$:IF
LP=1 THEN LPRINT XC;" ------ ";XC$:LPRINT
730 NEXT Z%
740 INPUT#1,XE%:IF XE%=0 PRINT "No enterPrise codes.":GOTO
800:ELSE PRINT XE%;" enterPrise codes.
750 IF LP=1 THEN CLS:PRINT"Ready printer for EnterPrise
codes then Press <ENTER>":PO%=500:A1%=0:GOSUB40130:PRINT
760 A$="MASTER ENTERPRISE CODE CATALOG":IF LP=1 THEN LPRINT
TAB(33-(LEN(A$)/2));A$:LPRINT
TAB(33-(LEN(A$)/2));STRING$(LEN(A$),"="):LPRINT:LPRINT
770 FOR Z%=1 TO XE%:INPUT#1,XF,XF$:PRINT XF;" ------ ";XF$:IF
LP=1 THEN LPRINT XF;" ------ ";XF$:LPRINT
780 NEXT Z%
```

```
800 INPUT#1,XG%:IF XG%=0 PRINT "No operator codes.":GOTO
820:ELSE PRINT XG%;" operator codes."
810 IF LP=1 THEN CLS:PRINT"Ready printer for Operator codes
then Press <ENTER>":PO%=500:A1%=0:GOSUB40130:PRINT
820 A$="MASTER OPERATOR CODE CATALOG":IF LP=1 THEN LPRINT
TAB(33-(LEN(A$)/2));A$:LPRINT
TAB(33-(LEN(A$)/2));STRING$(LEN(A$),"="):LPRINT:LPRINT
830 FOR Z%=1 TO XG%:INPUT#1,XH,XH$:PRINT XH;" ------ ";XH$:IF
LP=1 THEN LPRINT XH;" ------ ";XH$:LPRINT
840 NEXT Z%
850 CLOSE:PRINT:PRINT "End of CONFIG/DAT -- Press <ENTER>
":PO%=500:A1%=0:GOSUB40130:RUN
860 REM
870
REM=============================================================
880 REM            PREPARE TO ALTER CONFIG/DAT
890
REM=============================================================
900 GOTO 920
910 IF A$="Y" OR A$="y" THEN Z%=1:RETURN:ELSE IF A$="N" OR
A$="n" THEN Z%=0:RETURN:ELSE Z%=2:RETURN
920 CLS:PRINT "<< Alter CONFIG/DAT >>":PRINT:PRINT
930 PRINT"Answer either 'Y' or 'N' to the followin9
questions:"
940 PRINT:PRINT"Do you wish to alter the account codes?";
950 PO%=360:A1%=1:GOSUB40130:A$=AN$:PRINT:GOSUB 910:IF Z%=2
GOTO 950 ELSE IF Z%=1 THEN A1=1 ELSE A1=0
```

```
960 PRINT:PRINT"Do you wish to alter the bank/account
codes?";
970 PO%=493:A1%=1:GOSUB40130:A$=AN$:PRINT:GOSUB 910:IF Z%=2
GOTO 970 ELSE IF Z%=1 THEN A2=1 ELSE A2=0
980 PRINT:PRINT"Do you wish to alter the enterprise codes?";
990 PO%=619:A1%=1:GOSUB40130:A$=AN$:PRINT:GOSUB 910:IF Z%=2
GOTO 990 ELSE IF Z%=1 THEN A3=1 ELSE A3=0
1000 PRINT:PRINT"Do you wish to alter the operator codes?";
1010 PO%=745:A1%=1:GOSUB40130:A$=AN$:PRINT:GOSUB 910:IF Z%=2
GOTO 1010 ELSE IF Z%=1 THEN A4=1 ELSE A4=0
1015 IF A1=0 AND A2=0 AND A3=0 AND A4=0 THEN RUN ELSE OPEN
"O",2,"SCRATCH/DAT:"+DR$:OPEN "I",1,"CONFIG/DAT:"+DR$
1020 CLS
1030 PRINT"<< Alter account codes >>"
1040 PRINT:PRINT"Stand by... retrieving current account
codes"
1050 INPUT#1,XA%:DIM XD(XA%+10),XD$(XA%+10),XT(XA%+10):IF
XA%=0 GOTO 1065
1060 FOR Z%=1 TO XA%:INPUT#1,XD(Z%),XD$(Z%),XT(Z%):NEXT Z%
1065 IF A1=0 GOTO 1090
1070 CLS:PRINT"Do you wish to change or delete any existing
codes (Y/N)?"
1080 PO%=58:A1%=1:GOSUB40130:A$=AN$:PRINT:IF A$<>"Y" AND
A$<>"y" AND A$<>"N" AND A$<>"n" GOTO 1080 ELSE IF A$="Y" OR
A$="y" GOTO 1300 ELSE GOTO 1500
1090 PRINT#2,XA%:IF XA%=0 GOTO 1100 ELSEFOR Z%=1 TO
XA%:PRINT#2,XD(Z%);XD$(Z%);",";XT(Z%):NEXT Z%
```

```
1100 GOTO 2000
1300 FOR Z%=1 TO XA%:IF Z%>XA% GOTO 1490
ELSECLS:PRINT"Account code";Z%;" of";XA%
1310 PRINT"Account code # ";XD(Z%):PRINT"Description -
";XD$(Z%):PRINT"Account type - ";XT(Z%)
1320 PRINT:PRINT"Do you wish to:":PRINT:PRINT"1) Leave it as
it is.":PRINT"2) Change it.":PRINT"3) Delete
it" PRINT:PRINT"Enter your choice (1, 2, or 3)"
1330 PO%=735:A1%=1:GOSUB40130:A$=AN$:PRINT:IF A$<>"1" AND
A$<>"2" AND A$<>"3" GOTO 1330
1340 IF A$="1" GOTO 1490
1350 IF A$="3" GOTO 1400
1360 CLS:PRINT"Old account code - ";XD(Z%):PRINT "New
account code -
";:PO%=83:A1%=5:GOSUB40160:XD(Z%)=VAL(AN$):PRINT
1370 CLS:PRINT"Old account descr. - ";XD$(Z%):PRINT "New
account descr. -
";:PO%=256:A1%=50:GOSUB40130:XD$(Z%)=AN$:PRINT
1380 CLS:PRINT"Old account type - ";XT(Z%):PRINT "New
account type -
";:PO%=83:A1%=2:GOSUB40160:XT(Z%)=VAL(AN$):PRINT
1390 GOTO 1490
1400 REM
================================================================
1401 REM      MOVE ACCOUNTS DOWN A NOTCH AND DECREMENT XA%
1402 REM
================================================================
```

```
1410 XD(XA%+1)=0:XD$(XA%+1)="":XT(XA%+1)=0:FOR Q=Z% TO
XA%:XD(Q)=XD(Q+1):XD$(Q)=XD$(Q+1):XT(Q)=XT(Q+1):NEXT
Q:XA%=XA%-1:Z%=Z%-1
1490 NEXT Z%
1500 CLS:PRINT"Do you wish to add any new account codes
(Y/N)
1510 PO%=47:A1%=1:GOSUB40130:A$=AN$:PRINT:IF A$<>"Y" AND
A$<>"y" AND A$<>"N" AND A$<>"n" GOTO 1510 ELSE IF A$="N" OR
A$="n" GOTO 1090
1520 CLS:PRINT "How many new account codes do you wish to
add":PO%=46:A1%=4:GOSUB40160:NA=VAL(AN$):PRINT
1525 IF NA>10 PRINT "No more than ten additions at a time,
Please..":GOTO 1520
1530 FOR Z%=XA%+1 TO XA%+NA:CLS:PRINT"Account code";Z%;"
of";XA%+NA
1540 PRINT:PRINT "Enter account code
":PO%=147:A1%=5:GOSUB40160:XD(Z%)=VAL(AN$):PRINT
1550 PRINT:PRINT"Enter account code
description":PO%=320:A1%=50:GOSUB40130:XD$(Z%)=AN$:PRINT
1560 PRINT"Enter account type
":PO%=403:A1%=2:GOSUB40160:XT(Z%)=VAL(AN$):PRINT
1570 NEXT Z%:XA%=XA%+NA:GOTO 1090
2000 INPUT#1,XB%:CLS:PRINT"< Alter Bank/Account Codes >":DIM
XC(XB%+10),XC$(XB%+10):IF XB%=0 GOTO 2005 ELSE FOR Z%=1 TO
XB%:INPUT#1,XC(Z%),XC$(Z%):NEXT Z%
2005 IF A2=0 GOTO 2020
2010 PRINT"Do you wish to change or delete any existing
```

```
bank/account":PRINT"codes (Y/N)?"
2015 PO%=141:A1%=1:GOSUB 40130:A$=AN$:PRINT:IF A$<>"Y" AND
A$<>"y" AND A$<>"N" AND A$<>"n" GOTO 2015 ELSE IF A$="Y" OR
A$="y" GOTO 2030 ELSE GOTO 2500
2020 PRINT#2,XB%:IF XB%=0 GOTO 3000 ELSE FOR Z%=1 TO
XB%:PRINT#2,XC(Z%);XC$(Z%):NEXTZ%:GOTO 3000
2030 REM
2040 FOR Z%=1 TO XB%:IF Z%>XB% GOTO 2490
ELSECLS:PRINT"Bank/Account code";Z%;" of";XB%:PRINT
2042 PRINT"Bank/Account code -
";XC(Z%):PRINT:PRINT"Bank/Account code descr.
-":PRINTXC$(Z%):PRINT:PRINT"Do you wish to: ":PRINT:PRINT"1)
Leave it as it is.":PRINT"2) Change it.":PRINT"3) Delete
it."
2043 PRINT:PRINT"Enter your choice (1, 2, or 3)"
2045 PO%=863:A1%=1:GOSUB 40130:A$=AN$:PRINT:IF A$<>"1" AND
A$<>"2" AND A$<>"3" GOTO 2045 ELSE IF A$="1" GOTO 2490 ELSE
IF A$="3" GOTO 2400
2050 CLS:PRINT"Old bank/account code - ";XC(Z%):PRINT "New
bank/account code - ":PO%=88:A1%=5:GOSUB
40160:XC(Z%)=VAL(AN$):PRINT
2060 PRINT"Old bank/account code descr. -
":PRINTXC$(Z%):PRINT:PRINT"New bank/account code descr. -
":PO%=384:A1%=50:GOSUB 40130:XC$(Z%)=AN$:PRINT
2070 GOTO 2490
2400
REM==========
```

```
3410 REM  MOVE BANK/ACCOUNT CODES DOWN A NOTCH AND DECREMENT
 XB%
3420 REM
=============================================================
3430 XC(XB%+1)=0:XC$(XB%+1)="":FOR Q=Z% TO
XB%:XC(Q)=XC(Q+1):XC$(Q)=XC$(Q+1):NEXTQ:XB%=XB%-1:Z%=Z%-1
3490 NEXT Z%
2500 CLS:PRINT"Do you wish to add new bank/account codes
(Y/N)?"
2510 PO%=49:A1%=1:GOSUB 40130:A$=AN$:IF A$<>"Y" AND A$<>"y"
AND A$<>"N" AND A$<>"n" GOTO 2510 ELSE IF A$="N" OR A$="n"
GOTO 2020
2520 CLS:PRINT "How many new bank/account codes do you wish
to add":PO%=51:A1%=4:GOSUB40160:NA=VAL(AN$):PRINT
2525 IF NA>10 PRINT "No more than 10 additions at a time,
Please..":GOTO 2520
2530 FOR Z%=XB%+1 TO XB%+NA:CLS:PRINT "Bank/Account
code";Z%;" of";XB%+NA:PRINT
2540 PRINT "Enter bank/account code
":PO%=152:A1%=5:GOSUB40160:XC(Z%)=VAL(AN$):PRINT:PRINT:PRINT"
Enter bank/account
description":PO%=320:A1%=50:GOSUB40130:XC$(Z%)=AN$:PRINT:NEXT
Z%:XB%=XB%+NA:GOTO 2020
3000 CLS:PRINT"< Alter Enterprise Codes >":INPUT#1,XE%:DIM
XF(XE%+10),XF$(XE%+10):IF XE%=0 GOTO 3005 ELSE FOR Z%=1 TO
XE%:INPUT#1,XF(Z%),XF$(Z%):NEXTZ%
3005 IF A3=0 GOTO 3020 ELSE IF XE%=0 GOTO 3500
```

```
3010 PRINT:PRINT"Do you wish to change or delete existing
enterprise":PRINT"codes (Y/N)?"
3020 PO%=205:A1%=1:GOSUB 40130:A$=AN$:IF A$<>"Y" AND A$<>"y"
AND A$<>"N" AND A$<>"n" GOTO 3015 ELSE IF A$="y" OR A$="Y"
GOTO 3030 ELSE GOTO 3500
3030 PRINT#2,XE%:IF XE%=0 GOTO 4000 ELSE FOR Z%=1 TO
XE%:PRINT#2,XF(Z%);XF$(Z%):NEXT Z%:GOTO 4000
3040 FOR Z%=1 TO XE%:IF Z%>XE% GOTO 3490 ELSE
CLS:PRINT"EnterPrise code";Z%;" of";XE%:PRINT
3045 PRINT"EnterPrise code - ";XF(Z%):PRINT "EnterPrise
description:":PRINT XF$(Z%):PRINT:PRINT"Do you wish
to":PRINT:PRINT"1) Leave it as it is.":PRINT"2) Change
it.":PRINT "3) Delete it."
3047 PRINT:PRINT"Enter your choice (1, 2, or 3)"
3048 PO%=799:A1%=1:GOSUB 40130:A$=AN$:IF A$<>"1" AND A$<>"2"
AND A$<>"3" GOTO 3048 ELSE IF A$="1" GOTO 3490 ELSE IF
A$="3" GOTO 3300
3050 CLS:PRINT"Old enterprise code - ";XF(Z%):PRINT "New
enterprise code - ":PO%=86:A1%=5:GOSUB
40160:XF(Z%)=VAL(AN$):PRINT:PRINT:PRINT"Old enterprise
description -":PRINTXF$(Z%):PRINT"New enterprise description
-":PO%=384:A1%=50
3051 GOSUB 40130:XF$(Z%)=AN$:GOTO 3490
3300
REM=============================================================
3301 REM   MOVE ENTERPRISE CODES DOWN A NOTCH AND DECREMENT
XE%
```

```
3502
REM==========================================
3510 XF(XE%+1)=0:XF$(XE%+1)="":FOR Q=Z% TO
XE%:XF(Q)=XF(Q+1):XF$(Q)=XF$(Q+1):NEXT Q:XE%=XE%-1:Z%=Z%-1
3490 NEXT Z%
3500 CLS:PRINT"Do you wish to add any new enterPrise codes
(Y/N)?"
3510 PO%=51:A1%=1:GOSUB 40130:A$=AN$:IF A$<>"Y" AND A$<>"y"
AND A$<>"N" AND A$<>"n" GOTO 3510 ELSE IF A$="N" OR A$="n"
GOTO 3020
3520 CLS:PRINT "How many new enterPrise codes do you wish to
add":PO%=49:A1%=4:GOSUB40160:NA=VAL(AN$):PRINT
3530 IF NA>10 THEN PRINT "No more than 10 additions at a
time, Please..":GOTO 3520
3540 FOR Z%=XE%+1 TO XE%+NA:CLS:PRINT"EnterPrise code";Z%;"
of";XE%+NA
3550 PRINT:PRINT "EnterPrise code -
":PO%=146:A1%=5:GOSUB40160:XF(Z%)=VAL(AN$):PRINT:PRINT:PRINT"
EnterPrise code description -
":PO%=320:A1%=50:GOSUB40130:XF$(Z%)=AN$:PRINT:NEXT
Z%:XE%=XE%+NA:GOTO 3020
4000 CLS:PRINT"< Alter oPerator codes >":INPUT#1,XG%:DIM
XH(XG%+10),XH$(XG%+10):IF XG%=0 GOTO 4005 ELSE FOR Z%=1 TO
XG%:INPUT#1,XH(Z%),XH$(Z%):NEXT Z%
4005 IF A4=0 GOTO 4020 ELSE IF XG%=0 GOTO 4500
4010 PRINT:PRINT"Do you wish to change or delete any
existing codes (Y/N)?"
```

```
4013 PO%=186:A1%=1:GOSUB40130:A$=AN$:IF A$<>"Y" AND A$<>"y"
AND A$<>"N" AND A$<>"n" GOTO 4015 ELSE IF A$="N" OR A$="n"
GOTO 4500 ELSE GOTO 4030
4020 PRINT#2,XG%:IF XG%=0 GOTO 5000 ELSE FOR Z%=1 TO
XG%:PRINT#2,XH(Z%);XH$(Z%):NEXTZ%:GOTO 5000
4030 FOR Z%=1 TO XG%:IF Z%>XG% GOTO 4490 ELSE
CLS:PRINT"OPerator code";Z%;" of";XG%:PRINT:PRINT"OPerator
code - ";XH(Z%):PRINT"OPerator description -":PRINTXH$(Z%)
4035 PRINT:PRINT"Do you wish to:":PRINT:PRINT"1) Leave it as
it is.":PRINT"2) Change it.":PRINT"3) Delete it."
4037 PRINT:PRINT"Enter your choice (1, 2, or 3)"
4040 PO%=799:A1%=1:GOSUB 40130:A$=AN$:IF A$<>"1" AND A$<>"2"
AND A$<>"3" GOTO 4040 ELSE IF A$="1" GOTO 4490 ELSE IF
A$="3" GOTO 4300
4050 CLS:PRINT"Old operator code - ";XH(Z%):PRINT "New
operator code -
":PO%=85:A1%=5:GOSUB40160:XH(Z%)=VAL(AN$):PRINT:PRINT:PRINT"O
ld operator description - ":PRINTXH$(Z%):PRINT"New operator
description - "
4052 PO%=384:A1%=50:GOSUB40130:XH$(Z%)=AN$:PRINT:GOTO 4490
4300
REM=============================================================
4301 REM MOVE OPERATOR CODES DOWN A NOTCH AND DECREMENT XG%
4302
REM=============================================================
4310 XH(XG%+1)=0:XH$(XG%+1)="":FOR Q=Z% TO
XG%:XH(Q)=XH(Q+1):XH$(Q)=XH$(Q+1):NEXT Q:XG%=XG%-1:Z%=Z%-1
```

```
4490 NEXT Z%
4500 CLS:PRINT"Do you wish to add any new operator codes
(Y/N)?"
4510 PO%=49:A1%=1:GOSUB40130:A$=AN$:IF A$<>"Y" AND A$<>"y"
AND A$<>"N" AND A$<>"n" GOTO 4510 ELSE IF A$="N" OR A$="n"
GOTO 4020
4520 CLS:PRINT "How many new operator codes do you wish to
add":PO%=47:A1%=4:GOSUB40160:NA=VAL(AN$):PRINT:IF NA>10 THEN
PRINT "No more than 10 additions at a time, Please..":GOTO
4520
4530 CLS:FOR Z%=XG%+1 TO XG%+NA:CLS:PRINT"Operator
code";Z%;" of";XG%+NA:PRINT
4540 PRINT "Operator code -
":PO%=144:A1%=5:GOSUB40160:XH(Z%)=VAL(AN$):PRINT:PRINT:PRINT"
Operator code description -
":PO%=320:A1%=50:GOSUB40130:XH$(Z%)=AN$:PRINT:NEXT
Z%:XG%=XG%+NA:GOTO 4020
5000
REM=============================================================
5001 REM NOW PUT SCRATCH/DAT INTO CONFIG/DAT
5002
REM=============================================================
5005 CLS:PRINT"Finished with changes... writing back
CONFIG/DAT"
5010 CLOSE:OPEN "I",1,"SCRATCH/DAT:"+DR$:OPEN
"O",2,"CONFIG/DAT:"+DR$
5020 INPUT#1,XA%:PRINT#2,XA%:FOR Z%=1 TO
```

```
XA%:INPUT#1,XD(Z%),XD$(Z%),XT(Z%):PRINT#2,XD(Z%);XD$(Z%);",";
XT(Z%):NEXT Z%
5030 INPUT#1,XB%:PRINT#2,XB%:IF XB%=0 GOTO 5040 ELSE FOR
Z%=1 TO
XB%:INPUT#1,XC(Z%),XC$(Z%):PRINT#2,XC(Z%);XC$(Z%):NEXT Z%
5040 INPUT#1,XE%:PRINT#2,XE%:IF XE%=0 GOTO 5050 ELSE FOR
Z%=1 TO
XE%:INPUT#1,XF(Z%),XF$(Z%):PRINT#2,XF(Z%);XF$(Z%):NEXT Z%
5050 INPUT#1,XG%:PRINT#2,XG%:IF XG%=0 GOTO 5060 ELSE FOR
Z%=1 TO
XG%:INPUT#1,XH(Z%),XH$(Z%):PRINT#2,XH(Z%);XH$(Z%):NEXTZ%
5060 CLOSE:KILL "SCRATCH/DAT:"+DR$:RUN
6000 CLS:PRINT">> Sort CONFIG/DAT By Code
<<":PRINT:PRINT:PRINT"This sort will Place items in
ascending order":PRINT"according to their ";Z9$:PRINT:PRINT
6010 PRINT"Press '1' to start sorting or '2' to abort."
6020 A$=INKEY$:IF A$<>"1" AND A$<>"2" GOTO 6020 ELSE IF
A$="2" THEN RUN
6030 OPEN "I",1,"CONFIG/DAT:"+DR$:OPEN
"O",2,"SCRATCH/DAT:"+DR$
6040 PRINT">> Sorting Accounts <<":INPUT#1,XA:IF XA=0 GOTO
6170 ELSE DIM XD(XA),XD$(XA),XT(XA):FOR Z=1 TO
XA:INPUT#1,XD(Z),XD$(Z),XT(Z):NEXTZ
6050 N=XA:L=1
6060 L=L%2:IF L<N GOTO 6060
6065 L=INT((L-1)/2)
6070 IF L=0 GOTO 6140
```

```
6080 LI=N-L
6090 FOR P=1 TO LI:H=P
6100 K=H+L
6110 IF ZZ=0 AND XD(K)>XD(H) GOTO 6120 ELSE IF ZZ=1 THEN
X1$=XD$(K):X2$=XD$(H):GOSUB 7100:IF X1$>X2$ GOTO 6120
6115
G=XD(H):G$=XD$(H):G1=XT(H):XD(H)=XD(K):XD$(H)=XD$(K):XT(H)=XT
(K):XD(K)=G:XD$(K)=G$:XT(K)=G1:H=H-L:IF H>0 THEN 6100
6120 NEXT P
6130 GOTO 6065
6140
REM=============================================================
6150 REM XA IS SORTED - WRITE IT TO SCRATCH/DAT
6160
REM=============================================================
6170 PRINT#2,XA:IF XA=0 GOTO 6200 ELSE FOR Z=1 TO
XA:PRINT#2,XD(Z);XD$(Z);",";XT(Z):NEXT Z
6200 PRINT">> Sorting Bank/Accounts <<":INPUT#1,XB:IF XB=0
GOTO 6350 ELSE DIM XC(XB),XC$(XB):FOR Z=1 TO
XB:INPUT#1,XC(Z),XC$(Z):NEXT Z
6210 N=XB:L=1
6220 L=L*2:IF L<N GOTO 6220
6225 L=INT((L-1)/2)
6230 IF L=0 GOTO 6320
6240 LI=N-L
6250 FOR P=1 TO LI
6255 H=P
```

```
6260 K=H+L
6270 IF ZZ=0 AND XC(K)>XC(H) GOTO 6280 ELSE IF ZZ=1 THEN
X1$=XC$(K):X2$=XC$(H):GOSUB 7100:IF X1$>X2$ GOTO 6280
6275
G=XC(H):G$=XC$(H):XC(H)=XC(K):XC$(H)=XC$(K):XC(K)=G:XC$(K)=G$
H=H-L:IF H>0 THEN 6260
6280 NEXT P
6290 GOTO 6225
6320
REM=================================================
6330 REM XB IS SORTED - WRITE IT TO SCRATCH/DAT
6340
REM=================================================
6350 PRINT#2,XB:IF XB=0 GOTO 6400 ELSE FOR Z=1 TO
XB:PRINT#2,XC(Z);XC$(Z):NEXTZ
6400 PRINT">> Sorting EnterPrises <<":INPUT#1,XE:IF XE=0
GOTO 6550 ELSE DIM XF(XE),XF$(XE):FOR Z=1 TO
XE:INPUT#1,XF(Z),XF$(Z):NEXT Z
6410 N=XE:L=1
6420 L=L*2:IF L<N GOTO 6420
6425 L=INT((L-1)/2):IF L=0 GOTO 6520
6430 LI=N-L
6440 FOR P=1 TO LI:H=P
6445 K=H+L
6450 IF ZZ=0 AND XF(K)>XF(H) GOTO 6460 ELSE IF ZZ=1 THEN
X1$=XF$(K):X2$=XF$(H):GOSUB 7100:IF X1$>X2$ GOTO 6460
6455
```

```
G=XF(H):G$=XF$(H):XF(H)=XF(K):XF$(H)=XF$(K):XF(K)=G:XF$(K)=G$
:H=H-L:IF H>0 THEN 6445
6460 NEXT P
6470 GOTO 6425
6520
REM=============================================
6530 REM XE IS SORTED - WRITE IT TO SCRATCH/DAT
6540
REM=============================================
6550 PRINT#2,XE:IF XE=0 GOTO 6600 ELSE FOR Z=1 TO
XE:PRINT#2,XF(Z);XF$(Z):NEXT Z
6600 PRINT">> Sorting Operators <<":INPUT#1,XG:IF XG=0 GOTO
6730 ELSE DIM XH(XG),XH$(XG):FOR Z=1 TO
XG:INPUT#1,XH(Z),XH$(Z):NEXT Z
6610 N=XG:L=1
6620 L=L*2:IF L<N GOTO 6620
6625 L=INT((L-1)/2):IF L=0 GOTO 6720
6630 LI=N-L
6640 FOR P=1 TO LI:H=P
6645 K=H+L
6650 IF ZZ=0 AND XH(K)>XH(H) GOTO 6670 ELSE IF ZZ=1 THEN
X1$=XH$(K):X2$=XH$(H):GOSUB 7100:IF X1$>X2$ GOTO 6670
6660
G=XH(H):G$=XH$(H):XH(H)=XH(K):XH$(H)=XH$(K):XH(K)=G:XH$(K)=G$
:H=H-L:IF H>0 GOTO 6645
6670 NEXT P
6680 GOTO 6625
```

```
 6730
 REM=============================================================
 6730 REM XG IS SORTED - WRITE IT TO SCRATCH/DAT
 6740
 REM=============================================================
 6750 PRINT#2,XG:IF XG=0 GOTO 6800 ELSE FOR Z=1 TO
 XG:PRINT#2,XH(Z);XH$(Z):NEXT Z
 6800 CLS:PRINT"Finishing - writing back
 CONFIG/DAT":GOTO 5010
 7000 Z9$="descriptions":ZZ=1:GOTO 6000
 7100
 REM=============================================================
 7110 REM  REMOVE ALL LOWERCASE LETTERS FROM TEST STRING
 7120
 REM=============================================================
 7130 Z$="":FOR Z9=1 TO LEN(X1$):R$=MID$(X1$,Z9,1):IF
 ASC(R$)>90 THEN R$=CHR$(ASC(R$)-32)
 7140 Z$=Z$+R$:NEXT Z9:X1$=Z$
 7150 Z$="":FOR Z9=1 TO LEN(X2$):R$=MID$(X2$,Z9,1):IF
 ASC(R$)>90 THEN R$=CHR$(ASC(R$)-32)
 7160 Z$=Z$+R$:NEXT Z9:X2$=Z$:RETURN
40070 AN$="  ":POKE VARPTR(AN$),A1%:POKE
VARPTR(AN$)+2,INT(PO%/256)+60:POKE
VARPTR(AN$)+1,PO%-INT(PO%/256)*256:RETURN
40130 Q8%=PO%:Q9%=A1%:A%=0:PRINT@PO%,STRING$(A1%,132);
40131  IF A%=A1%THEN40134ELSEPRINT@PO%+A%,CHR$(132);
40132 A$=INKEY$:IFA$=""THEN40132ELSEIFINSTR("
```

```
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcd
efghijklmnopqrstuvwxyz",A$)THENPRINT@PO%+A%,A$;:A%=A%+1:GOTO4
0134
40133
IF(INSTR(CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40135,40130,40138:GO
TO40131
40134 A$=INKEY$:IF A$=""THEN40134ELSE40133
40135 IFA%<A1%THENPRINT@PO%+A%,CHR$(132);
40136 A%=A%-1:IFA%<0THENA%=0:GOTO40131ELSE40131
40137 A%=0
40138
IFA$=CHR$(91)THENPRINT@PO%,STRING$(A1%,132);ELSEPRINT@PO%+A%,
STRING$(A1%-A%," ");
40139 GOSUB40070:GOTO51000
40140 S%=1:AN$="":PRINT@PO%,"$";STRING$(A1%,132);"
";:PRINT@PO%+A1%-2,".";
40141
A$=INKEY$:IFA$=""THEN40141ELSEIFINSTR("0123456789",A$)THEN401
43ELSEIFINSTR("-"+CHR$(8)+CHR$(31)+CHR$(13),A$)GOTO40145,4014
4,40148,40147
40142 GOTO40141
40143
AN$=AN$+A$:IFLEN(AN$)=1THENAN$=CHR$(132)+AN$ELSEIFLEN(AN$)>A1
%-1THENAN$=LEFT$(AN$,A1%-1)ELSEIFLEN(AN$)=3ANDLEFT$(AN$,1)=CH
R$(132)THENAN$=RIGHT$(AN$,2)
40144
PRINT@PO%+A1%-LEN(AN$),LEFT$(AN$,LEN(AN$)-2);".";RIGHT$(AN$,2
```

```
);GOTO40141
 40145
 S%=-S%:PRINT@PO%+A1%+1,"";:IFS%=-1THENPRINT"-";:GOTO40141ELSE
 PRINT" ";:GOTO40141
 40146 IF A$=CHR$(91)THENPRINT@PO%+1,STRING$(A1%,132);"
 ";:PRINT@PO%+A1%-2,".";:GOTO40149ELSEPRINT@PO%,STRING$(A1%+2,
 " ");:GOTO40149
 40147 IFLEN(AN$)=0THENPRINT@PO%,STRING$(A1%+2,"
 ");:GOTO40149 ELSEPRINT@PO%+1,STRING$(A1%-1-LEN(AN$),"
 ");:IFLEFT$(AN$,1)=CHR$(132)THENPRINT@PO%+A1%-1,"0";:MID$(AN$
 ,1,1)="0"
 40148
 AN$=MID$(AN$,1,LEN(AN$)-2)+"."+RIGHT$(AN$,2):IFS%=-1THENAN$="
 -"+AN$
 40149 GOSUB 40170:RETURN
 40160 S%=1:AN$="":PRINT@PO%,STRING$(A1%,132);" ";
 40161
 A$=INKEY$:IFA$=""THEN40161ELSEIFINSTR("0123456789",A$)THEN401
 62ELSEONINSTR(CHR$(8)+CHR$(31)+"."+"-"+CHR$(13),A$)GOTO40160,
 40160,40165,40163,40166:GOTO40161
 40162
 AN$=AN$+A$:IFLEN(AN$)>A1%THENAN$=LEFT$(AN$,A1%):GOTO40161ELSE
 PRINT@PO%+A1%-LEN(AN$),AN$;:GOTO40161
 40163
 S%=-S%:PRINT@PO%+A1%,"";:IFS%=-1THENPRINT"-";ELSEPRINT" ";
 40164 GOTO40161
 40165 IFINSTR(AN$,".")=0THEN40162ELSE40161
```

```
40166 IFAN$=""THEN40168ELSEPRINT@PO%,STRING$(A1%-LEN(AN$),"

40167 IFS%=-1THENAN$="-"+AN$:GOTO40169ELSE40169

40168 IF A$=CHR$(91)THENPRINT@PO%,STRING$(A1%,132);"

ELSEPRINT@PO%,STRING$(A1%," ");" ";

40169 GOSUB 40170:RETURN

40170 AN#=VAL(AN$):AN#=(INT((AN#+.005)*100)/100):RETURN

50999 REM * String Compression subroutine

51000 IF LEN(AN$)=0 THEN 51030 ELSE IF MID$(AN$,1,1)=" "

THEN AN$=RIGHT$(AN$,LEN(AN$)-1):GOTO 51000

51010 IF LEN(AN$)=0 THEN 51030 ELSE IF

MID$(AN$,LEN(AN$),1)=" " THEN AN$=LEFT$(AN$,LEN(AN$)-1):GOTO

51010

51030 IF (LEN(AN$)=0 AND Q9%<>0) THEN PO%=Q8%:A1%=Q9%:GOTO

40130

51040 RETURN
```

Program Listing: DEPREC/BAS

```
0 REM * deprec/bas version 1 08/10/82
10 CLS:CLEAR10000:FI$="$$#,###,###.##":PRINT">> Option: Post
& Print Out Depreciation Record <<":PRINT
20 PRINT"Do you wish to:":PRINT:PRINT"1) Post a depreciation
entry":PRINT"2) Print out stored entries":PRINT"3) Abort and
return to MASTER MENU":PRINT:PRINT"Enter your choice (1, 2,
or 3)"
30 PO%=543:A1%=1:GOSUB40160:A$=AN$:IF A$<>"1" AND A$<>"2"
AND A$<>"3" GOTO 30
40 IF A$="3" THEN CLS:PRINT"Returning to MASTER MENU":RUN
"MASTER/BAS"
50 IF A$="2" GOTO 420
60 CLS:PRINT"Place the disk marked SPECIAL in Drive 1, then
Press <ENTER>"
70 PO%=64:A1%=0:GOSUB40130
80 OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:CLOSE:IF
D$="SPECIAL" GOTO 90 ELSE CLS:PRINT"ERROR - this is not a
SPECIAL disk!  It is marked: ";D$
85 PRINT@960,"Press <ENTER> to
continue";:PO%=1020:A1%=0:GOSUB40130:RUN
90 DIM ST(14):OPEN "I",1,"HEADER/DAT:1":INPUT#1,D$:FOR Z=1
TO 14:INPUT#1,ST(Z):NEXT Z:CLOSE
100 CLS:PRINT"Entry #";ST(8)+1:PRINT
110 PRINT "Enter a description (up to 25 chars.)"
120 PO%=192:A1%=25:GOSUB40130:DE$=AN$
```

```
120 PRINT:PRINT:PRINT"Enter the date this item was acquired
(up to 8 chars.)"
140 PO%=384:A1%=8:GOSUB40130:DA$=AN$:PRINT
150 CLS:PRINT"Select the category this item is
in":PRINT:PRINT"1) Buildings & Improvements":PRINT"2)
Machinery & Equipment":PRINT"3) Purchased Breeding
Livestock":PRINT:PRINT"Enter your choice (1, 2, or 3)"
160 PO%=415:A1%=1:GOSUB40160:TY=VAL(AN$):IF TY<>1 AND TY<>2
AND TY<>3 GOTO 160
170 CLS:PRINT"Enter the unrecovered cost - beginning of
year"
180 PO%=64:A1%=10:GOSUB40140:UC#=AN#:PRINT
190 PRINT"Enter the depreciation this year"
200 PO%=192:A1%=10:GOSUB40140:DY#=AN#:PRINT
210 CLS:PRINT"Verify that what you typed is
correct:":PRINT:PRINT"Description: ";DE$
220 PRINT:PRINT"Date acquired: ";DA$
230 PRINT:PRINT"Category: ";:IF TY=1 THEN PRINT "Buildings &
Improvements" ELSE IF TY=2 THEN PRINT "Machinery &
Equipment" ELSE IF TY=3 THEN PRINT "Purchased Breeding
Livestock"
240 PRINT:PRINT"Unrecovered cost - beginning of year:
";:PRINT USING FI$;UC#
250 PRINT:PRINT"Depreciation this year: ";:PRINT USING
FI$;DY#
260 PRINT@960,"Is this information: (1=correct  or
2=incorrect)";
```

```
        51010
    ON  40133 40141 40161
  OPEN  320 400 510 530 560 600 715
 CLOSE  320 400 520 530 560 600 800
LPRINT  720/3 730 740/3 750 760 770/3 780/3 800/3 810/3
        820/2 830 840/2
  POKE  40070/3
 PRINT  10/2 20/8 30 40 60/3 90/5 100/6 110 140/3 150/2
        160/3 170/2 180/3 190/2 200 210/2 220/2 230/2
        240/2 250 280/8 290 300/2 310 330/8 340 420/7
        430/4 440 500 510/# 520/# 530/#2 540/2 550 570/8
        580 610 620/6 630 700 710 2000 40130 40131 40132
        40135 40138/2 40140/2 40144 40145/3 40146/3 40147/3

        40160 40162 40163/3 40166 40168/2
 CLEAR  10
  TAB(  720/2 750 760 770 780 810 820 830 840
    TO  80 400 510 520 530 600
 USING  220 230 770 780 810 820 840
VARPTR  40070/3
STRING$ 720 730 740 800 830 40130 40138/2 40140 40146/2
        40147/2 40160 40166 40168/2
 INSTR  40132 40133 40141/2 40161/2 40165
INKEY$  40132 40134 40141 40161
  THEN  40 60 90 120 135 260 290 340 440 580 630 40131
        40132/2 40134 40135 40136 40138 40141/2 40143/3
        40145 40146 40147/2 40148 40161/2 40162 40163
```

```
      40165 40166 40167 40168 51000/2 51010/2
  +   520/2 740 40070/3 40131 40132/2 40133/2 40135
      40138 40140 40141/3 40143/2 40144 40145/2 40146/3
      40147/3 40148/3 40161/4 40162/2 40163 40167 40170
  -   440 720/2 840 40070 40136 40138 40140 40143/2
      40144/2 40145/2 40146 40147/3 40148/2 40162 40163/2

      40166 40167 51000 51010
  *   40070 40170
  /   720/2 40070/2 40170
 AND  30/2 110 250 290 340 440 580 630 40143
  OR  30 110 250 290 340 440 580 630
  >   30/3 110/2 135 250/2 290/2 340/2 440/2 580/2
      630/2 40143 40162
  =   10 30/4 40 50 60/4 80 90/4 110/4 120/2 135 140/3
      150/3 160/3 170/3 250/4 260 290/6 310/2 320 340/6
      400 410 440/6 510 520/6 530/2 550/2 560 580/6
      600 610 630/6 710/2 720 730 740 840 850 2000/2
      40070 40130 40131 40132/3 40134/2 40136/2 40137
      40138 40140/2 40141/2 40143/7 40145/2 40146 40147/3

      40148/3 40160/2 40161/2 40162/2 40163/2 40165
      40166 40167/2 40168 40170/2 51000/3 51010/3
  <   30/3 60 110/2 250/2 290/2 340/2 440/2 580/2 630/2
      40135 40136
 INT  40070/2 40170
 LEN  90 135 720/3 740 40143/3 40144/2 40147/2 40148
```

	40162/2 40166 51000/2 51010/3
VAL	60 140 150 40170
CHR$	40131 40133/3 40135 40138 40141/3 40143/2 40146
	40147 40161/3 40168
LEFT$	135 40143/2 40144 40147 40162 51010
RIGHT$	40143 40144 40148 51000
MID$	40147 40148 51000 51010

Cross Reference of Command Keywords: SUMMARY/BAS

```
  FOR  30 40 50 60 90 120 150 180 210 360 370 380 390
       400 440 460 480 500 520 580 590 610 630 710 722/2
       830 850 870 880 970 990 1000 1100 1200 1210
  CLS  10 65 85 99 130 160 190 220 230 240 260 270 290
       330 340 350 800 802 1010 1030 1040 1080 1100
       1120 1200 1220 2010
 NEXT  30 40 50 60 90 120 150 180 210 370 380 390 400
       420 440/2 460 465 480 485 500 505 550 580 600
       620 640 710 720 722 723 724 725 850 880 890 920
       970 990 1000 1100 1200 1210
 DATA  860
INPUT  30/#2 40/#2 50/#2 60/#2 90/# 330/# 370/#2 380/#2
       390/#2 400/#2 1060/# 1100/#2
  DIM  25 30 40 50 60 62 90 1100
 READ  850
 GOTO  40 50 60 100 110 120/2 130 140 150/2 160 170
       180/2 190 200 210/2 220 250 270 280 320/2 330
       340 380 390 400 420 440/2 460/2 465 480/2 485
       500/2 505 530/3 540/2 570 580/3 590 600 610 620
       630 640 710 720 722 725 800 840 1020 1060 1080
       1090 1100 1120 2010/2 40132 40133/2 40136 40141
       40142 40144 40145/2 40146/2 40147 40161/2 40162/2
       40164 40167
  RUN  260 1030 1080 1120 1220
   IF  40 50 60 110/2 120/2 140 150/2 170 180/2 200
```

```
        210/2 250 260 270 280 320/2 330 340 355 380 390
        400 420 440/2 460/2 480/2 500/2 530/2 540 570
        580/2 590/2 610/2 630/2 710 722 800/2 870 880
        1020 1030 1060 1080/2 1100 1120/2 2010 40131
        40132/2 40134 40135 40136 40138 40141/2 40143/3
        40145 40146 40147/2 40148 40161/2 40162 40163
        40165 40166 40167 40168
RESTORE 850
  GOSUB 80 100 110 120 140 150 170 180 200 210 250 320
        800 810 840/2 1020 1050 1080 1120 2010 40139
        40149 40169
 RETURN 115 850 40070 40139 40149 40169 40170
    REM 0 410 412 414 430 432 434 450 452 454 470 472
        474 490 492 494 510 512 514 560 562 564 700 702
        704 721 50000 50001 50002 50003 50004 50005 50006
        50007 50008
   ELSE 40 50 60 110/2 120/2 140 150/2 170 180/2 200
        210/2 320 355 380 390 400 440 460/2 480/2 500/2
        530/2 540 580/2 590 610 630 710 800 1060 1080
        1100 1120 2010 40131 40132 40134 40136 40138
        40141/2 40143/2 40145 40146 40147 40161/2 40162
        40163 40165 40166 40167 40168
     ON 40133 40141 40161
   OPEN 30 90 330 355/2 1060 1100 1200 1210
  CLOSE 65 90 240 330 1060 1100 1200 1210
 LPRINT 820/2 830/3 840/3 870/5 880/5 890/4 900/2 910/2
        930 940/4 950 960 970/6 980 990/2 1000/3
```

```
   POKE  40070/3
  PRINT  10 20/2 70 80 85 100 110 120/2 130 140 150/2
         160 170 180/2 190 200 210/2 220 240/8 250 260
         270 290 300 310 320/2 330 340 350 360 800 802
         810 1010 1030 1040 1060 1070/7 1080 1100/3 1110/8
         1120 1200 1200/# 1210/#2 1220 2000 40130 40131
         40132 40135 40138/2 40140/2 40144 40145/3 40146/3
         40147/3 40160 40162 40163/3 40166 40168/2
  CLEAR  10
   TAB(  350 820/2 870 880 950/3 960/3 970/3 980 990 1000
     TO  30 40 50 60 90 120 150 180 210 360 370 380 390
         400 440 460 480 500 520 580 590 610 630 710 722/2
         830 850 870 880 970 990 1000 1100 1200 1210
  USING  870 880 890 900 910 970/3 990 1000/2
 VARPTR  40070/3
STRING$  820 830 930 40130 40138/2 40140 40146/2 40147/2
         40160 40166 40168/2
  INSTR  40132 40133 40141/2 40161/2 40165
 INKEY$  40132 40134 40141 40161
   THEN  110 120 140 150 170 180 200 210 260 270 320 330
         340 355 420 440 530 580 590 610 630 722 800 870
         880 1030 1080 1120 40131 40132/2 40134 40135
         40136 40138 40141/2 40143/3 40145 40146 40147/2
         40148 40161/2 40162 40163 40165 40166 40167 40168
     +   110 590 610 630 715 723 870 880 990 1000 40070/3
         40131 40132/2 40133/2 40135 40138 40140 40141/3
         40143/2 40144 40145/2 40146/3 40147/3 40148/3
```

40161/4 40162/2 40163 40167 40170

\- 100/4 120 140 150 170 180 200 210 440 460 480
500 570/4 580 590 610 630 820/2 910 970 1000
40070 40136 40138 40140 40143/2 40144/2 40145/2
40146 40147/3 40148/2 40162 40163/2 40166 40167

* 110 40070 40170

/ 820/2 40070/2 40170

AND 110/3 250/2 270 320 330 570/3 800/3 870/2 880/2
1020/3 1080 1120 40143

OR 110/2 250 320 530/2 540 800/2 870 880 1030

> 110/4 250/3 320/2 330/2 440 530 540 722 800/4
870/2 880/2 1020/4 1080/2 1120/2 40143 40162

= 10 30 40/2 50/2 60/2 80/2 90 100/5 110/8 120/7
140/3 150/7 170/3 180/7 200/3 210/7 230 250/4
260 270/3 280 320/5 330 340/2 355 360 370 380/2
390/2 400/2 420 440/2 460/3 480/3 500/3 520/2
530/2 570/4 580/4 590/5 610/5 630/5 710/2 715
722/2 723 800/7 810/2 820 830 840/2 850 870/3
880/3 910 970/2 990/3 1000/4 1020/3 1030/2 1050/2
1060 1080/4 1100/2 1120/4 1200 1210/2 2010/4
40070 40130 40131 40132/3 40134/2 40136/2 40137
40138 40140/2 40141/2 40143/7 40145/2 40146 40147/3

40148/3 40160/2 40161/2 40162/2 40163/2 40165
40166 40167/2 40168 40170/2

< 110/4 250/3 320/2 330/2 440 530 540 722 800/4
870/2 880/2 1020/4 1080/2 1120/2 40135 40136

```
   INT  40070/2 40170
   LEN  820/3 40143/3 40144/2 40147/2 40148 40162/2 40166
   VAL  120 150 180 210 40170
  CHR$  40131 40133/3 40135 40138 40141/3 40143/2 40146
        40147 40161/3 40168
 LEFT$  40143/2 40144 40147 40162
RIGHT$  40143 40144 40148
  MID$  40147 40148
```

BIBLIOGRAPHY

American Society Farm Managers and Rural Appraisers. Professional Farm And Ranch Management Manual. Denver, 1977.

Anderson, Jock R., Dillon, John L., and Hardaker, J. Brian. Agricultural Decision Analysis. Ames: Iowa State University Press, 1977.

Agrawal, R. C., and Heady, Earl O. Operations Research Methods For Agricultural Decisions. Ames: Iowa State University Press, 1972.

Barnard, C. S., and Nix, J. S. Farm Planning And Control. London: Cambridge University Press, 1973.

Castle, Emery N., Becker, Manning H., and Smith, Frederick J. Farm Business Management. New York: Macmillan Publishing Co., Inc., 1972.

Efferson, J. Norman. Principles Of Farm Management. New York: McGraw-Hill Book Company, Inc., 1953.

Finkel, Leroy, and Brown, Jerald R. Data File Programming in BASIC. New York: John Wiley & Sons, Inc., 1981.

Forster, D. Lynn, and Erven, Bernard L. Foundations For Managing The Farm Business. Columbus: Grid Publishing, Inc., 1981.

Hopkins, John A., and Murray, William G. Elements Of Farm Management. New York: Prentice-Hall, Inc., 1953.

Hudelson, Robert R. Farm Management. New York: The Macmillan Company, 1939.

James, Sydney C., and Stoneberg, Everett. Farm Accounting Business Analysis. Ames: Iowa State University Press, 1974.

University of Missouri - Columbia, College of Agriculture - Extension Division. Farm Planning Handbook. Columbia, MO: University of Missouri Press, 1978.